Titan2 series of products HMIC_S IP User Guide

（UG052001, V1.7）

（2022-09-01）

Shenzhen Ziguang Tongchuang Electronics Co., Ltd.

Document Version Revision History

Date	Document version	Revision history	Applicable IP and corresponding version
2020-12-31	V1.0	1. Initial release.	V1.0
2021-01-12	V1.1	1. Detailed description of PHYdebug interface.	V1.1
2021-01-20	V1.1a	1. Adjust the PHYdebug interface, delete the control interface for sending and receiving prbs code after eye diagram calibration, and add CK and DQS signal delay adjustment ports.	V1.1a, V1.1b, V1.1c, V1.1d
2021-03-05	V1.2	1. PHY interface arrangement; 2. Add Ping Pong PHY description; 3. The description of the IP interface has been updated.	V1.2
2021-04-09	V1.2a	1. The PHY interface is sorted and updated.	V1.2a
2021-04-30	V1.3	1. Adjust the manual structure and standardize the format; 2. The document version revision record adds the "applicable IP and corresponding version" description column; 3. Add IP version change records; 4. Add "2.3.5 Resource Utilization"; 5. Update the Demo board model in the Example Design system block diagram in the "2.4.1 Design Block Diagram" section; 6. Add "2.6.2 Register Access"; 7. Updated the block diagram of the IP internal clock structure in the chapter "2.8.3 Clock Constraints"; 8. Add "2.9 IP debugging method".	V1.3
2021-07-14	V1.3e	1. Update the applicable device package information, see "2.1.2 Applicable Devices and Packages"; 2. Update the Modelsim simulation script file name corresponding to ExampleDesign, see "2.3.1 3 Generate IP", "2.3.3 1 Modelsim Simulation"; 3. Update the ExampleDesign system block diagram, see "2.4.1 Design Block Diagram"; 4. Added a detailed comparison between SimplifiedAXI4 and standard AXI4, see "2.5.2 1 SimplifiedAXI4 interface"; 5. Improve the Debug interface signal classification, see "2.5.3 4 Debug Interface"; 6. Improve the Debug data description, see "2.5.5 Debug data description"; 7. For a detailed comparison between the newly added DFI interface and the standard DFI, see "2.5.4 1 DFI Interface Specification"; 8. Added typical applications in three scenarios of single BANK-x16, dual BANK-x32, and multiple BANK-x64, see "2.7.1 Typical application of single BANK-x16", "2.7.2 Typical application of dual BANK-x32","2.7.3 Typical application of multiple BANK-x64"; 9. Delete wiring constraints and add related content of physical constraints, see "2.8.5 Physical constraints";	V1.3e

Date	Document version	Revision history	Applicable IP and corresponding version
		10. Added dqsgateupdate function description, see "2.8.6 dqs gate update function description"; 11. Add the reference document [12] UG050010_Titan2 series HMIC_S IP application guide, and improve the relevant instructions of the manual; 12. Add the IP version change record corresponding to V1.3e; 13. Updated the glossary of terms and abbreviations, see "3.2 Glossary" and "3.3 List of acronyms".	
2021-08-13	V1.3f	1. Update the IP selection interface, see "2.3.11 Select IP"; 2. Update the screenshot of ModelSim execution simulation script, see "2.3.3 1 Modelsim simulation"; 3. Correct the footnote format of the table, see Table 2-3 and Table 2-15; 4. Correct the packet length description related to the read and write timing of the address channel of the SimplifiedAXI4 interface, see "2.5.21(1) Write address channel timing", "2.5.21(2) Read address channel timing"; 5. Correct the debug_data interface bit width in the normal debugging output signal, see Table 2-19; 6. Update the debug_calib_ctrl interface bit width in common debugging output signals, and update the definition and description of related fields synchronously, see Table 2-23; 7. Updated the debug interface description, added 5 debug interfaces (debug_rd_dq_dly_bin, debug_rd_dbi_dly_bin, debug_rdvref_mode, dbg_dram_vref_range, dbg_dram_vref_value), see Table 2-19; 8. Added a description of the default value of the IP interface configuration parameters, see "2.3.12 Configuring IP Parameters".	V1.3f
2021-08-18	V1.4	1. Update resource utilization data, see "2.3.5 Resource Utilization"; 2. Update the IP selection interface, see "2.3.11 Select IP"; 3. Add the IP version change record corresponding to V1.4.	V1.4
2021-09-15	V1.4a	1. Add the IP version change record corresponding to the V1.4a version; 2. Regulate document version revision records and IP version change record formats; 3. Add PDS version requirements, see "2.3.1 Module Instantiation"; 4. Improve the description of Pin/BankOption and BasicOptions parameters, see Table 2-2 and Table 2-4; 5. Complete the list of output files after IP generation, see Table 2-5.	V1.4a
2021-09-30	V1.4b	1. Add the IP version change record corresponding to V1.4b version; 2. Improve the description of module instantiation, update the screenshot of the IP selection path interface, see "2.3.1 Module instantiation"; 3. Correct the default values of DebugSignals related parameters, see Table 2-2;	V1.4b

Date	Document version	Revision history	Applicable IP and corresponding version
		4. Correct the cross-reference links related to the generated IP path description, see "2.3.1 3 Generate IP"; 5. Correct the packet length description related to the read and write timing of the SimplifiedAXI4 interface address channel, see "2.5.2 1(1) Write Address Channel Timing", "2.5.2 1(2) Read Address Channel Timing"; 6. Refer to Table 2-19 to update the port names of the debugging interface (dbg_dram_vref_left_edge_phy0, dbg_dram_vref_ margin_win_phy0, dbg_dram_vref_value_phy0); 7. Update the meaning description of the error_status debug port related fields in debug_calib_ctrl, see Table 2-23.	
2021-10-28	V1.4c	1. Add the IP version change record corresponding to the V1.4c version; 2. Update the text table style; 3. Update the name of the text specification list, see Table 1-1; 4. Update the format of the configuration parameter description list, and add the "IP configuration interface default value" column, see Table 2-2, Table 2-3, and Table 2-4; 5. Update the screenshot of the Basic Options page, and update the default values of Clock Frequency and EnableHardFifo configuration parameters, see Figure 2-6 and Table 2-2; 6. Update the screenshot of the Pin/Bank Options page, add the Enable fdc file select configuration parameter, and update the default values of the EnablePAR, EnableALERT_n, and EnableTEN configuration parameters, see Figure 2-8 and Table 2-4; 7. Improve the output file description after IP generation, see Table 2-5; 8. Update the maximum resource utilization table data, see Table 2-6.	V1.4c
2021-11-15	V1.4d	1. Add the IP version change record corresponding to V1.4d version; 2. Improve the description of the "Enablefdcfileselect" option, and support the selection of fdc files through the dialog box, see Table 2-4; 3. Update the maximum resource utilization table data, see Table 2-6; 4. Update the debugging interface list, delete the original read-write manual window scanning port, see Table 2-19.	V1.4d
2021-11-26	V1.4e	1. Add the IP version change record corresponding to the V1.4e version; 2. Update the list of debug interfaces, and add read and write window scanning enable ports (wr_win_scan_en, wr_win_scan_group_en, rd_win_scan_en, rd_win_scan_group_en), see Table 2-19; 3. Improve the DFI interface list on the Controller side and delete redundant ports (dfi_reset_n, dfi_init_complete, dfi_error), see Table 2-13; 4. Added instructions related to the debugging interface, see "Notes".	V1.4e

Date	Document version	Revision history	Applicable IP and corresponding version
2021-12-31	V1.5	1. Add the IP version change record corresponding to V1.5; 2. Update the description of the maximum data rate of DDR3/DDR4, see "2.1.1 Main Features", Table 2-2; 3. Update PDS version requirements, see "Note"; 4. Update the IP interface block diagram, see Figure 2-4; 5. Update the Basic Options page configuration interface, add the option area "Memory Address Mapping Selection" and related options, and update the configuration IP parameter interface screenshots, Basic Options page screenshots, Basic Options page configuration parameter description list, see Figure 2-5, Fig. 2-6, Table 2-2; 6. Update the resource utilization list and add resource utilization related to DDR4 Controller + PHY mode and DDR3 x64 configuration mode, see Table 2-6; 7. Improve the Global interface list and add ZQCS related ports (ddr_zqcs_req, ddr_zqcs_ack), see Table 2-7; 8. Update the debug interface list, add the debug signals related to group data alignment errors in the read direction (align_error_clr, align_error, align_error_lck), and improve the description of the output port of the IOB output impedance calibration value, see Table 2-19; 9. Add DDR4 Mode Register related description, see "2.6.1 1 Controller Internal Register"; 10. Delete the original "Column Address Rounding Example" section.	V1.5
2022-01-21	V1.5a	1. Add the IP version change record corresponding to the V1.5a version; 2. Update the parameter description of Operating Mode, only PHY Only mode is supported, see Table 2-2; 3. Update the resource utilization list and delete the content related to the Controller + PHY mode, see Table 2-6. 4. For the description of legality check when adding CA pin constraints, see "Note".	V1.5a
2022-05-31	V1.6	1. Add the IP version change record corresponding to V1.6; 2. Update the software version description, see "Note"; 3. Update the configuration interface of the Basic Options page, add the option area "DBI/DM Options" and related options, and update the screenshots of the configuration IP parameter interface, the screenshots of the Basic Options page and the list of configuration parameter descriptions, and the screenshots of the Summary page, see Figure 2-5, Figure 2-6, Table 2-2, Figure 2-9; 4. Update the parameter description of Operating Mode, add support for Controller + PHY mode, see Table 2-2; 5. Update the parameter description of Memory Part, add support for MT40A512M8XX particle type, see Table 2-3;	V1.6

Date	Document version	Revision history	Applicable IP and corresponding version
		6. Update the relevant description of the original Simplified AXI4 interface to the standard AXI4 interface, delete the detailed difference list between the original Simplified AXI4 and the standard AXI4, and update the AXI4 interface description and timing description synchronously, see "2.5.1 2 AXI4 Interface", "2.5.2 1 AXI4 interface"; 7. Update the description of port debug_cpd_offset_dir, see Table 2-17; 8. Add the description of simulation acceleration, see "2.3.3 Running Simulation". 9. Add the description of the read data channel ready signal in the AXI4 interface, see "2.5.1 2(4) Read Data Channel", "2.5.2 1(4) Read Data Channel Timing".	
2022-07-28	V1.6a	1. Add the IP version change record corresponding to the V1.6a version; 2. Update the configuration interface of the Basic Options page, add the option area "Rescal" and related options, and update the screenshot of the configuration IP parameter interface, the screenshot of the Basic Options page and the list of configuration parameter descriptions, and the screenshot of the Summary page, see Figure 2-5 and Figure 2- 6. Table 2-2, Figure 2-9;	V1.6a
2022-09-01	V1.7	1. Add the IP version change record corresponding to V1.7; 2. Update the version description of the software, see "Note".	V1.7

IP Version Change History

IP Version	Update description	Release time
V1.0	1. PG2T HMIC_S IP initial version.	2020-12-21
V1.0	1. Organize the debug interface; 2. Open particle DLL by default; 3. Fix the problem that the MEM_BADDR_WIDTH parameter of the ddr_top layer is incorrect when instantiating IP; 4. Unencrypt several modules that are important for debugging; 5. Add FPGA vref voltage manual configuration function.	2020-12-31
V1.0	1. Adjust the TERM_OFF_SET and TERM_ON_SET parameters of GTP_OSERDES_E2 to solve the problem of reading and writing a single burst8 data error on the client-side dfi interface; 2. After modification, the board test of the demo board ddr4_1066x16 project is normal.	2021-01-01
V1.1	1. Add dfi_prbs module for dfi interface testing; 2. Adjust the term timing of dq and dqs; 3. Logic timing optimization; 4. Fix several bugs in IP; 5. When IP generates 1866 projects, the default parameters are optimized; 6. The UI interface is fully functional; 7. Fix dqsp/dqsn waveform asymmetry problem.	2021-01-16
V1.1a	1. Add delay to CK and CKN output, and the delay value can be configured through the port; 2. Fix the problem of inaccurate calculation in the write calibration module window; 3. Fix the timing problem of dfi interface; 4. The delay adjustment function is added to the dqs clock in the read direction, and the delay can be increased or decreased through port configuration.	2021-01-20
V1.1b	1. Optimize the read and write window, and realize that the reference design vcc can work normally without pressurization; 2. Add the testparam timing optimization parameter file to the reference design.	2021-01-.22
V1.1c	1. 1. The DDR3 function can be used normally; 2. 2. Optimize the reading calibration algorithm; 3. 3. Optimize the write calibration algorithm; 4. 4. UI interface function optimization.	2021-01-27
V1.1d	1. Optimize the timing of ddrphy code; 2. Fix the problem that some dq write windows will jump when the client resets multiple times at low temperature.	2021-02-02
V1.2	1. Add hardfifo enable option; 2. Combine Ping-Pong PHY into IP; 3. Added the function of manually adjusting the window in the reading direction; 4. PHY timing optimization; 5. Optimize the clock structure and reduce the utilization of GPLL resources; 6. Reference design constraint optimization; 7. Optimized and updated the IP configuration interface; 8. The dqs gate algorithm is updated.	2021-03-10

IP Version	Update description	Release time
V1.2a	1. DFI interface bist function update; 2. Perfect IP function; 3. PHY timing optimization; 4. Fixed several bugs.	2021-04-09
V1.3	1. Timing optimization; 2. Perfect reference design; 3. Change the low level of memory reset signal from 200us to 700us; 4. dqs gate algorithm optimization; 5. Modify the speed level range of CWL and CL parameters on the UI interface; 6. Optimize the IO constraint impedance value; 7. Add the same delay value to the CA signal.	2021-04-30
V1.3e	1. Support skipping one or more training steps, and force training results through the port; 2. Support manual scan of write data window; 3. Optimized dqs gate update function; 4. When a training error occurs, the exact error steps can be reported through the error status port.	2021-07-19
v1.3f	1. Equipment test function bug fix; 2. Read vref training scheme optimization; 3. Some bug fixes for ddr3 controller.	2021-08-13
V1.4	1. Realize resource optimization on the basis of 1.3f; 2. The read vref training optimization not included in 1.3f.	2021-08-18
V1.4a	1. Incorporate into vref training optimization in 1.3f; 2. Debug port optimization, add group grouping mode; 3. UG update.	2021-09-15
V1.4b	1. Fix the problem that the interface configuration changes after the idf file is reloaded; 2. Fix the problem that the left and right sides of the writing window are asymmetrical; 3. UG update.	2021-09-30
V1.4c	1. Resource optimization in PP_PHY mode, two PHYs share the read-write calibration module; 2. Add optional training 1bit function; 3. Added the function of importing FDC; 4. The default reference design can run through the flow; 5. Fix XA-581; 6. Fix the problem of rdvref fine training failure when opening dm and reading dbi; 7. When DM calibration is enabled, the problem that DM is closed after training is completed; 8. Partial logic timing optimization; 9. UG update.	2021-10-28
V1.4d	1. Optimization of reading and writing calibration resources; 2. Delete the original read-write manual window scanning port, and use the force port to achieve the same function; 3. Write the calibration window information, simplify the debug port, and only keep the window information of bit0; 4. Training process optimization; 5. UG update.	2021-11-13

IP Version	Update description	Release time
V1.4e	1. Add the function of writing vref fine calibration; 2. Add read-write window scan enable port; 3. Code optimization, delete invalid logic; 4. UG update.	2021-11-26
V1.5	1. DDR4 supports Controller+PHY mode; 2. Added address mapping option in Controller+PHY mode; 3. Improve AXI BIST and DFI BIST functions, and support DDR3 and DDR4 at the same time; 4. Optimize DDR4 RTP parameter configuration; 5. Fix the problem that Controller read and write commands cannot cross ROW and BANK; 6. The highest rate of DDR4 supports up to 2000Mbps; 7. The reference design adds some internal signal muticycle constraints; 8. Improve debug information; 9. UG update.	2021-12-31
V1.5a	1. Repair DDR3 800M board, emulation is not normal; 2. Perform legality check when IP manually constrains CA pins; 3. Fix the problem of asymmetry between the left and right windows of the sampling point when reading the eye diagram and manually scanning; 4. Fix the problem that the data bit width on the interface changes when reloading idf after generating IP; 5. Limit the Operating Mode to PHY Only mode; 6. UG update.	2022-01-21
V1.6	1. Fix the problem that the synthesis file in the IP index file cannot be flexibly adapted to different modes; 2. Added DBI and DM configuration options on the UI interface; 3. Added support for DDR4 X8 particle types on the UI interface; 4. Add training result printing in simulation; 5. Fixed the problem that training failed when write calibration was disabled; 6. Fixed the problem of training failure when using DDR3 force rdcal; 7. Controller is upgraded from Simplified AXI4 interface to standard AXI4 interface; 8. Improve DM signal DDR3 as OUTPUT type, DDR4 as INOUT type; 9. The enable_fifo signal is divided according to the group, and the enable_fifo signal in the same bank is restricted to a fixed area; 10. UG update.	2022-05-31
V1.6a	1. When adding DDR3 on the interface, you can choose whether to use the RESCAL function; 2. Fix the problem of probabilistic initialization of DDR3 800M; 3. When the Controller+PHY mode is restricted on the interface, the data bit width can only be selected as 8, 16, 32, 64; 4. UG update.	2022-07-28
V1.7	2 When adapting to write leveling, the initial delay of ck is less than the initial delay of dqs; 3 Repair possible risks in the case of discontinuous read data validity; 4 UG update.	2022-09-01

Table of contents

Table of Contents

Figure catalog

Chapter 1 Preface

This chapter describes the scope of application of this manual, the structure of the manual and related writing rules, helping users quickly find the required information.

About this manual

This manual is a user guide for HMIC_S (High performance Memory Interface Controller Soft core) IP, a DDR4 IP launched by Tsinghua Unigroup. The IP is implemented based on the HPIO resources of Titan2 series FPGA products and is compatible with DDR3. The content of this manual mainly includes IP usage guide and related appendices. Through this manual, users can quickly understand the relevant features and usage methods of HMIC_S IP.

Specifications of the manual

Table 1-1 Specifications for writing

Text	Usage Principles
Attention	If the user ignores the attention content, it may cause certain adverse consequences or fail to operate successfully due to misoperation.
Description	Instructions and tips provided to the user.

Chapter 2 IP Usage Guide

This chapter describes the related usage guide of HMIC_S IP, including IP introduction, IP block diagram, IP generation process, Example Design, IP interface description, IP register description, typical application, instructions and precautions, and IP debugging methods. More detailed information about the design process can be found in the following PDS help documents.

- *Pango_Design_Suite_Quick_Start_Tutorial[1]*
- *Pango_Design_Suite_User_Guide[2]*
- *IP_Compiler_User_Guide[3]*
- *Simulation_User_Guide[4]*

2.1 Introduction to IP

HMIC_S IP is a DDR4 IP launched by Tsinghua Unigroup. It is based on the HPIO resources of Titan2 series FPGA products to realize SDRAM reading and writing. It is compatible with DDR3 and can be completed by the IPC (IP Compiler) tool in the company's PDS (Pango Design Suite) suite. Configure and generate.

2.1.1 Main Features

The main features of the HMIC_S IP product are as follows.

- Support DDR3, DDR4, DDR4 ping-pong PHY;
- Support a maximum data bit width of 72bit (Ping-Pong PHY maximum 32bit);
- User interface: AXI4 bus interface, APB bus interface;
- Support configurable low power consumption modes: Self-Refresh and Power Down;
- Support the highest data rate of DDR3 up to 1866Mbps;
- Support the highest data rate of DDR4 up to 2000Mbps;
- Burst Length 8 and single Rank;
- PHY can be used independently.

2.1.2 Applicable devices and packages

Table 2-1 HMIC_S IP Applicable Devices and Packages

Applicable Device	Package Type Support
PG2T390H	FFBG900

2.2 IP Block Diagram

The block diagram of the HMIC_S IP system is shown in Figure 2-1. HMIC_S IP provides two modes of Controller + PHY and PHY Only.

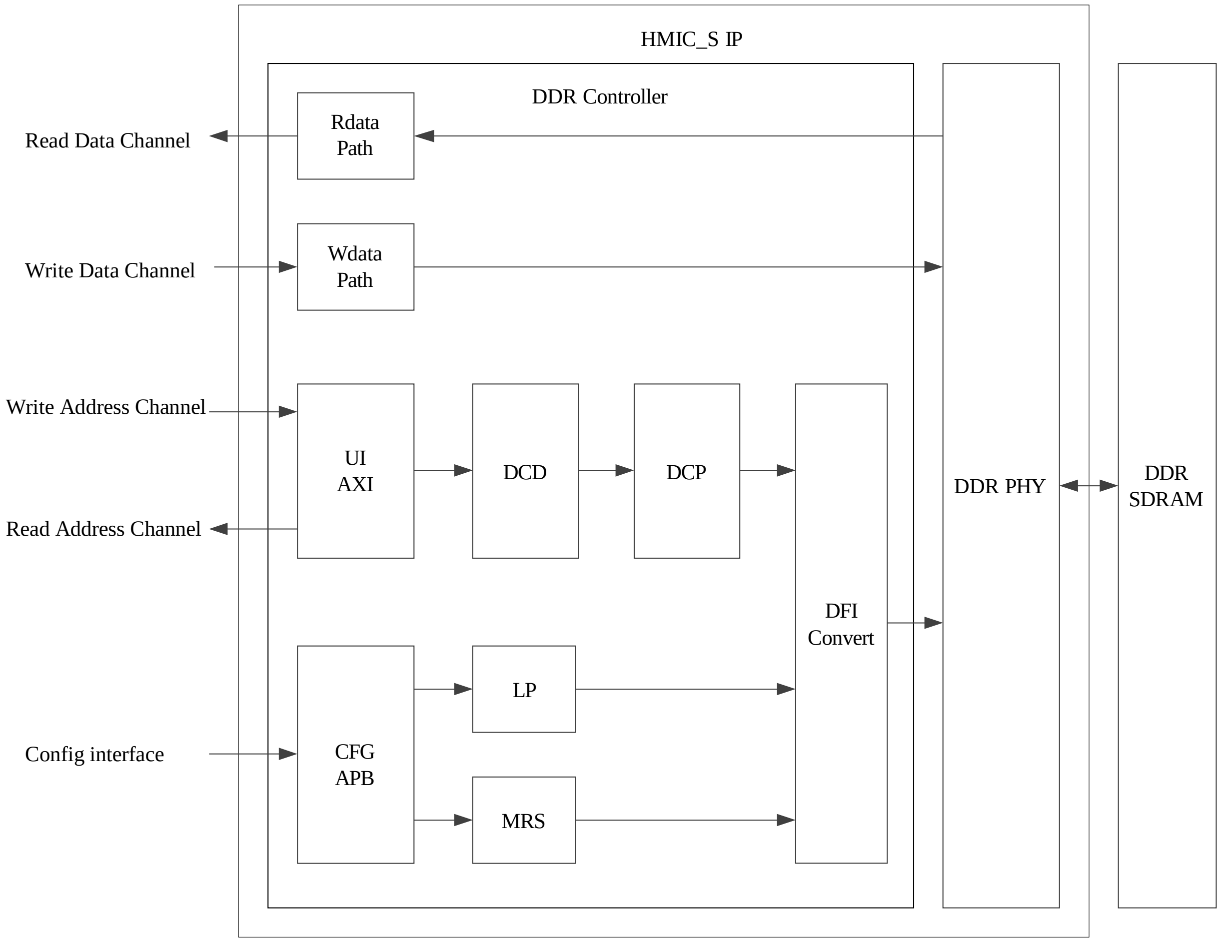


Figure 2-1 HMIC_S IP system block diagram

2.2.1 Controller + PHY

In this mode, the IP includes the functions of DDR Controller and DDR PHY. Users can read and write data through the AXI4 interface, and realize low power consumption and MRS control through the APB interface.

- AXI4 interface

The interface includes four parts: write address channel, read address channel, write data channel and read data channel.

The user initiates read and write operations through the write address channel and the read address channel; the commands are parsed into Controller internal commands in the UI AXI module; decomposed into DDR corresponding commands in the DCD (DDR Command Decode) module;

Implement DDR-based timing control in the DCP (DDR3 Command Procedure) module; convert it into a DFI interface in the DFI Convert module and pass it to the PHY, and finally pass it to the DDR Memory interface.

The write data is directly passed to the DDR PHY through the write data channel interface through the Wdata Path module, and finally passed to the DDR Memory interface.

The read data from DDR Memory is sampled and analyzed by DDR PHY, synchronized by the Rdata Path module, and returned to the user through the read data channel interface.

- Config Interface

This interface is an APB configuration interface, through this interface, users can read the status of DDR SDRAM to realize low power consumption and MRS control.

2.2.2 PHY Only

In this mode, users need to realize the design of the Controller by themselves, and interface with the PHY through the DFI interface to realize the operation of the DDR SDRAM.

2.3 IP generation process

2.3.1 Module Instantiation

The customized configuration of HMIC_S IP can be completed through the IPC tool, and the required IP modules can be instantiated. For the specific usage of IPC tools, please refer to IP_Compiler_User_Guide[3].

The main operation steps of HMIC_S IP module instantiation are described as follows.

1. Select IP

Open IPC, click File->Update in the main window to open the Update IP dialog box, and add the IP model of the corresponding version.

After selecting the device type of FPGA, the Catalog interface can display the loaded IP model. Select the corresponding version of DDR4 Interface under the System/DDR/Soft directory, and the IP selection path interface is shown in Figure 2-2. Then set the Pathname and Instance Name on the right page, and the project instantiation interface is shown in Figure 2-3.

Attention:

Software must be version 2022.1 and above.

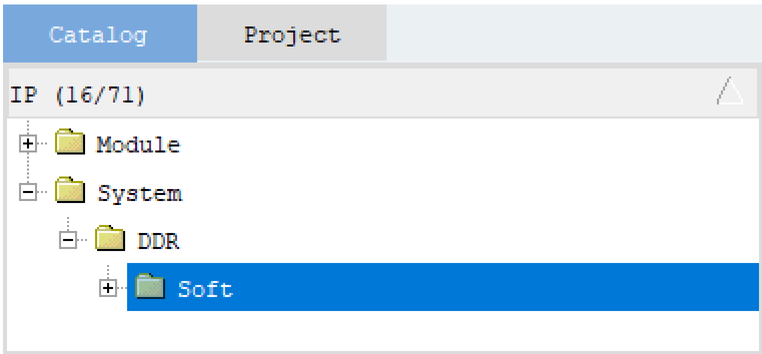


Figure 2-2 HMIC_S IP selection path interface

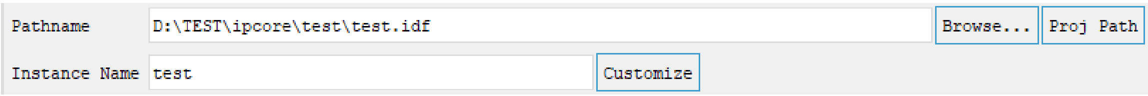


Figure 2-3 Project instantiation interface

2. Configure IP parameters

After the IP is selected, click <Customize> to enter the HMIC_S IP parameter configuration interface. The Symbol on the left is the interface block diagram, as shown in Figure 2-4; the right is the parameter configuration window, as shown in Figure 2-5.



Figure 2-4 HMIC_S IP interface block diagram

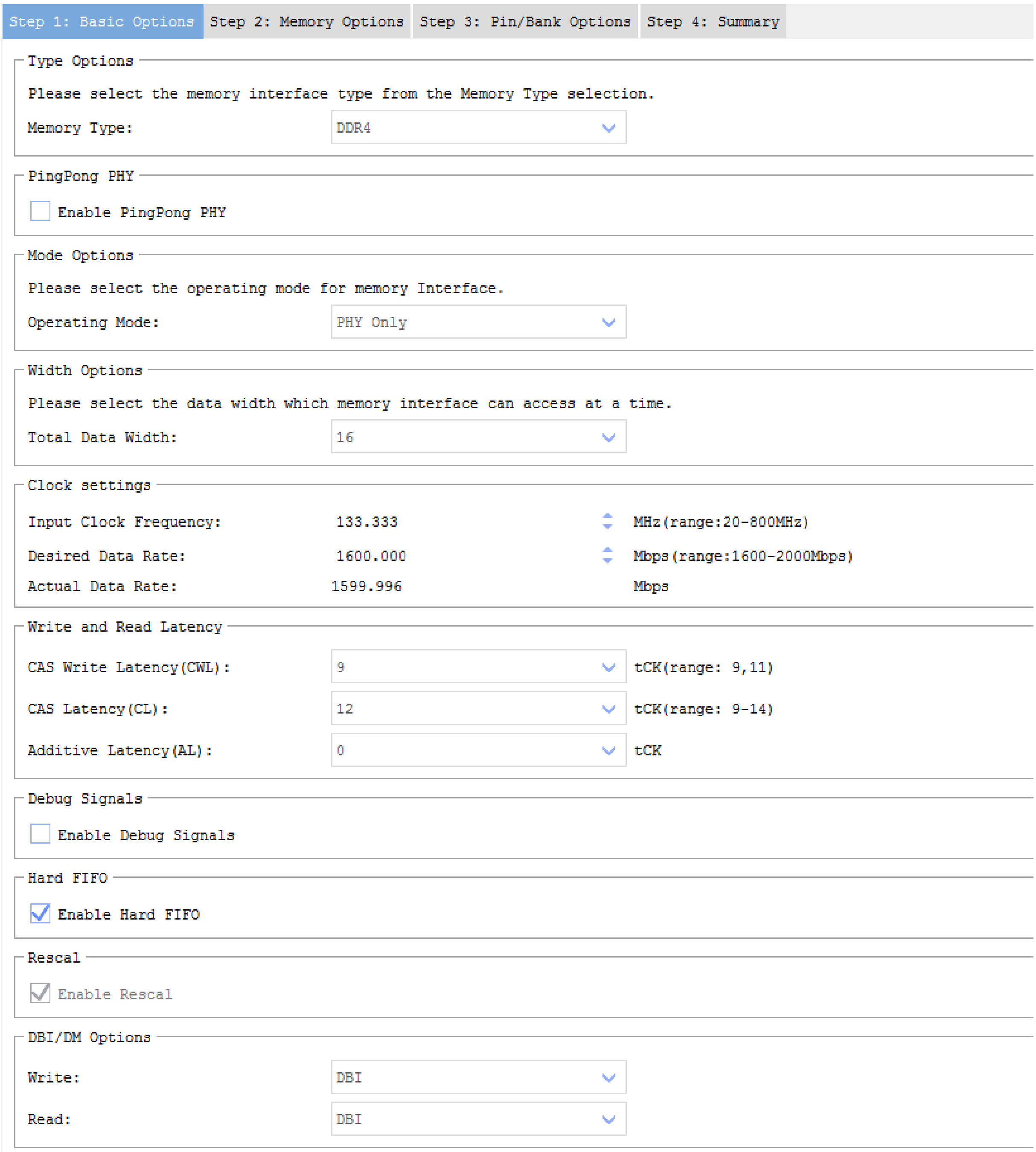


Figure 2-5 Configure HMIC_S IP parameter interface

Attention:

Please make sure to configure the IP parameters according to the page order, that is, Step 1 → Step 2 → Step 3 → Step 4.

The parameter configuration is divided into four pages, namely Step1: Basic Options, Step2: Memory Options, Step3: Pin/Bank Options, and Step4: Summary. The HMIC_S IP configuration steps are described below.

(1) Step 1: Basic Options

Basic Options is the basic configuration page of IP, the interface is shown in Figure 2-6, and the detailed parameters are shown in Table 2-2.

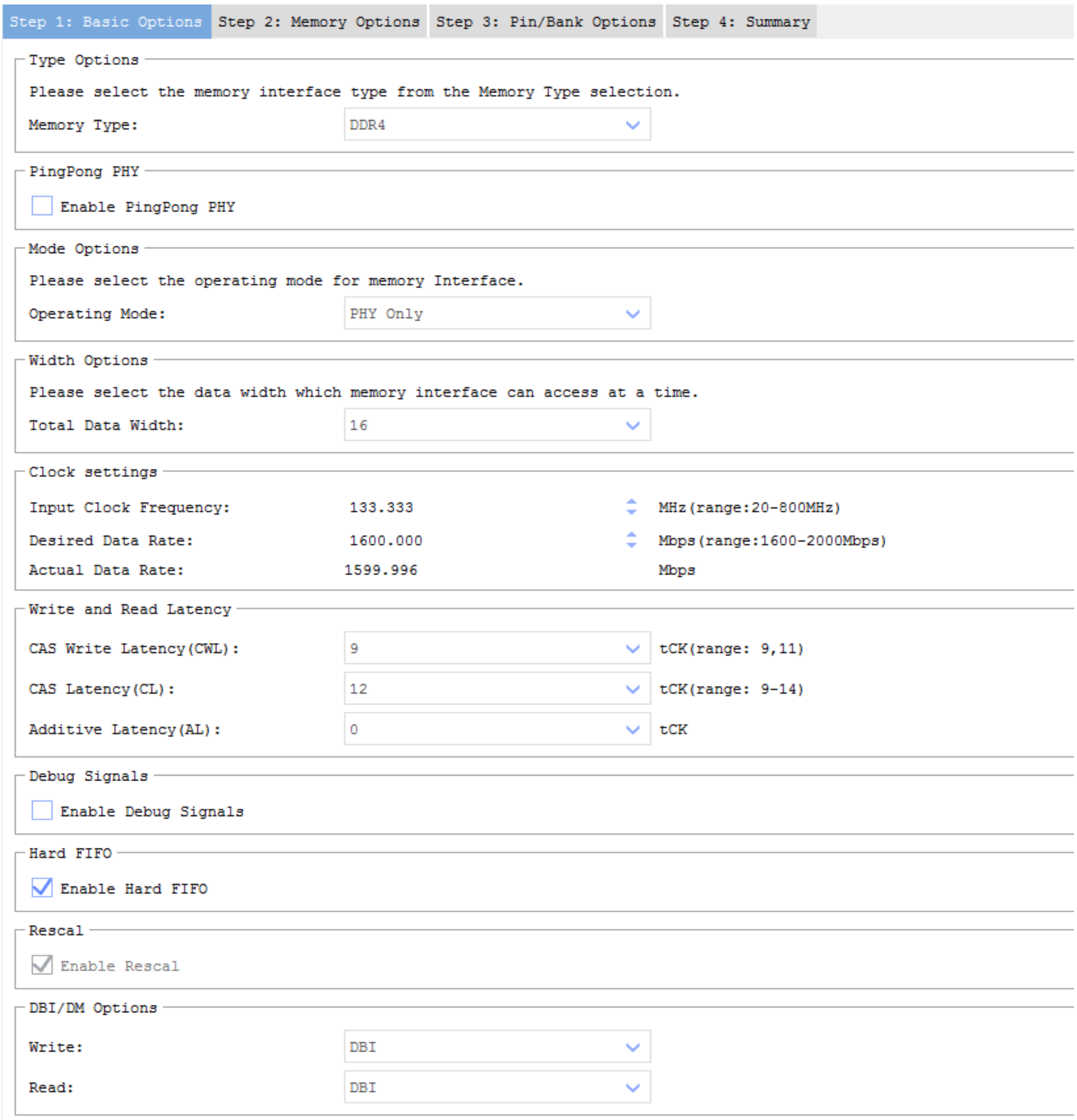


Figure 2-6 Basic Options page

Table 2-2 Description of configuration parameters on the Basic Options page

Options area	Parameters/configuration options	Parameter Description	IP configuration interface default value
Type Options	Memory Type	The type of SDRAM used, the currently supported types are: 1) DDR4; 2) DDR3.	DDR4
PingPong PHY	Enable PingPong PHY	DDR4 ping-pong PHY enable selection. Can be selected when Memory Type is DDR4.	Not Enabled
Mode Options	Operating Mode	HMIC_S Operating mode selection. The available operating modes are: 1) PHY Only; 2) Controller + PHY.	PHY Only
Memory Address Mapping Selection[1]	ROW + BANK + COLUMN	Select the read-write address mapping method of the Controller AXI interface as: "ROW + BANK + COLUMN", see Figure 2-16.	Checked
	BANK + ROW + COLUMN	Select the reading and writing address mapping method of the Controller AXI interface: "BANK + ROW + COLUMN", see Figure 2-17.	Unchecked
Width Options	Total Data Width[2]	Total DQ width of off-chip SDRAM connected to HMIC_S. The total widths currently supported are: 1) 72; 2) 64; 3) 56; 4) 48; 5) 40; 6) 32; 7) 24; 8) 16; 9) 8.	16
Clock settings	Input Clock Frequency	The input clock of HMIC_S, the unit is MHz.	133.333
	Desired Data Rate	Desired data rate in Mbps. DDR3 supports a maximum rate of 1866Mbps. DDR4 supports a maximum rate of 2000Mbps.	1600
	Actual Data Rate	The actual achievable data rate, as close as possible to the desired rate.	-
Write and Read Latency	CAS Write Latency(CWL)	CAS Write Latency configuration, the unit is tCK.	9
	CAS Latency(CL)	CAS Latency configuration, the unit is tCK.	12
	Additive Latency(AL)	Additive Latency configuration, unit tCK.	0
Debug Signals	Enable Debug Signals	Debug signal enable configuration.	Not Enabled
Hard FIFO	Enable Hard FIFO	Hardware circuit FIFO enable selection between IO Serdes and Fabric.	Enabled
Rescal	Enable Rescal[3]	RESCAL is enabled.	Enabled

[1] This option area is only displayed in "Controller + PHY" mode.
[2] In Controller+PHY mode, the data bit width can only be 8, 16, 32, 64.
[3] When the Memory Type is configured as DDR4, Rescal is enabled by default; when the Memory Type is configured as DDR3, Rescal is supported.

Options area	Parameters/configuration options	Parameter Description	IP configuration interface default value
DBI/DM Options	Write	Write direction configuration.	DBI
	Read	Read direction configuration.	DBI

Note: "-" indicates that the IP configuration interface has no default value for this parameter.

(2) Step 2: Memory Options

Memory Options is the configuration page of Memory parameters, the interface is shown in Figure 2-7, and the detailed parameters are shown in Table 2-3.

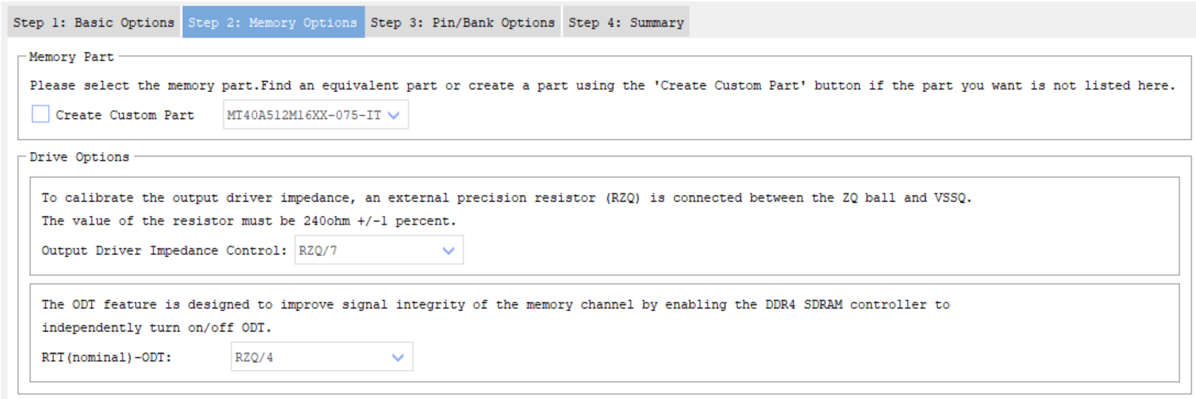


Figure 2-7 Memory Options page

Table 2-3 Description of configuration parameters on the Memory Options page

Options area	Parameters / configuration options	Parameter Description	IP configuration interface default value
Memory Part	SDRAM device model	DDR3 supported models are: 1） MT41J128M8XX-15E; 2） MT41J64M16XX-15E; 3） MT41J256M8XX-15E; 4） MT41J128M16XX-15E; 5） MT41J512M8XX-15E; 6） MT41J256M16XX-15E; DDR4 supported models are: 1） MT40A512M16XX-075-IT; 2） MT40A512M8XX.	MT40A512M16XX-075-IT
	Create Custom Part	Customize new SDRAM type enable selection. If the above device models supported by the IP do not meet the requirements, the user can check this option and customize a new one based on the reference model in the Custom Memory Part option box below. SDRAM type.	Unchecked

Options area	Parameters / configuration options	Parameter Description	IP configuration interface default value
Custom Memory Part[4]	Select Base Part	Reference model for custom SDRAM.	MT40A512M16XX- 075-IT
	Timing Parameters	Customize the timing parameters of SDRAM, including: trfc, tras, trp, trcd, twr, trefi, trtp, twtr.	trfc: 350 tras: 35 trp: 12.5 trcd: 12.5 twr: 15 trefi: 7.8 trtp: 7.5 twtr: 7.5
	Row Address	Row address.	16
	Column Address	Column address.	10
	Bank Address	Bank address.	2
	Bank Groups	Bank Groups address (DDR4 only).	1
Drive Options	Output Driver Impedance Control	Drive capability options. For details, please refer to: JESD79-3D, DDR3 SDRAM Standard[9], JESD79-4B, DDR4 SDRAM Standard[10].	RZQ/7
	RTT(nominal) - ODT		RZQ/4

(3) Step 3: Pin/Bank Options

Pin/Bank Options is the interface parameter configuration page, the interface is shown in Figure 2-8, and the detailed parameters are shown in Table 2-4.

Attention:

All configuration items in "Step 3: Pin/Bank Options" must be configured according to the pin assignment on the actual board. After generating the IP, constrain the pin positions of the DQ signal, reset signal, and status signal according to the pin assignment on the actual board, otherwise errors may occur when running Flow.

---

[4] This option box is displayed when the "Create Custom Part" option is checked, and it is used to customize a new SDRAM type.

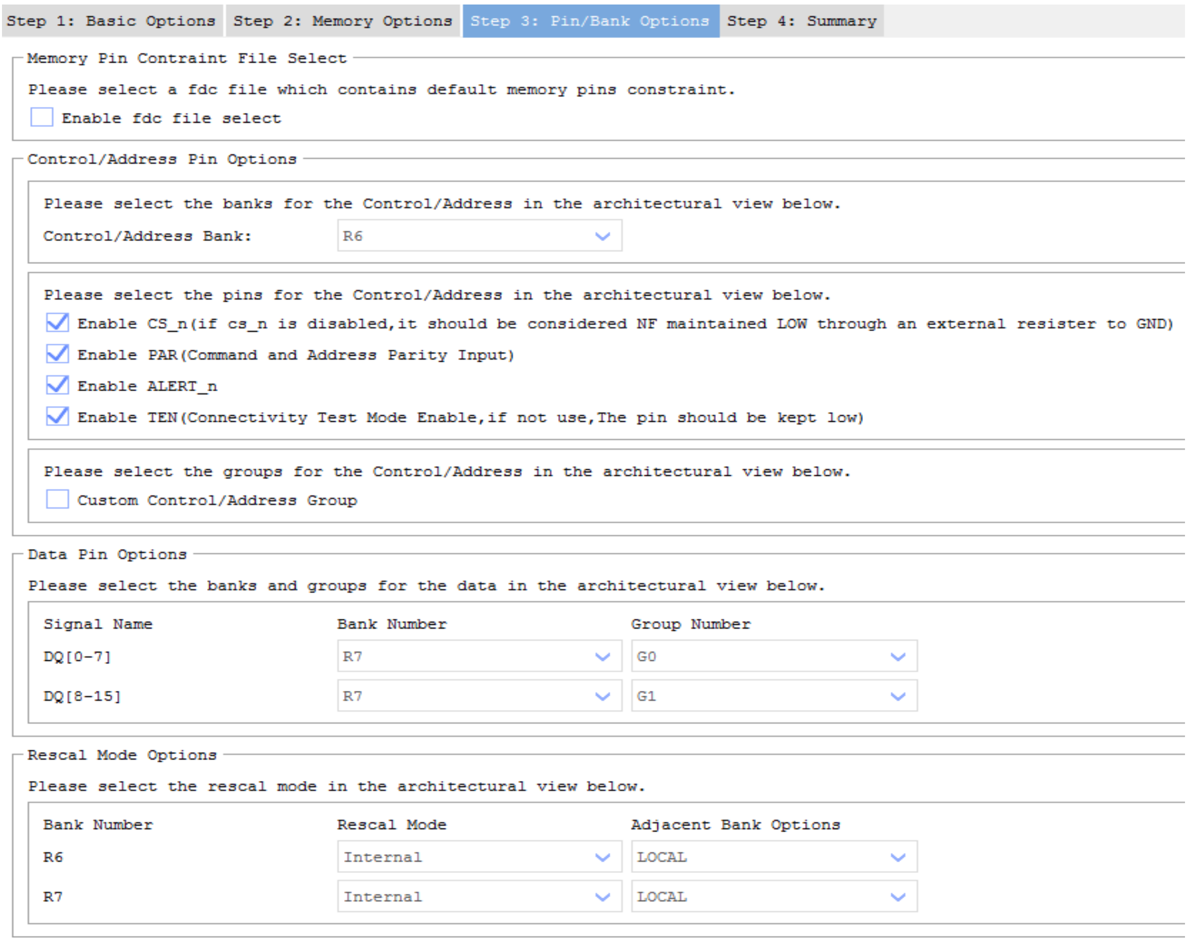


Figure 2-8 Pin/Bank Options page

Table 2-4 Description of Pin/Bank Options page configuration parameters

Options area	Parameters / configuration options	Parameter Description	IP configuration interface default value
Memory Pin Contraint File Select	Enable fdc file select	Custom fdc files can be imported. When enabled, you can enter the path of the user's fdc file in the text box, or select the fdc file through the dialog box, and the Memory interface constraints in the fdc file will be automatically read, and the Control/Address and Data Pin in the UI will be configured. Note: The signal name of the Memory interface in user fdc must be consistent with the signal name of the Memory interface in Example Design.	Not Enabled
Control / Address Pin Options	Control/ Address Bank	The bank where the control and address lines of the Memory interface are located	R6
	Enable CS_n	mem_cs_n signal enable selection.	Enabled
	Enable PAR	mem_parity signal enables selection.	Enabled
	Enable ALERT_n	mem_alert_n signal enable selection.	Enabled
	Enable TEN	mem_ten signal enables selection.	Enabled

Options area	Parameters / configuration options	Parameter Description	IP configuration interface default value
	Custom Control / Address Group	User-defined control and address bus group enable selection. When checked, the user can customize the grouping and pin assignment of each PAD, and when not checked, the default grouping will be adopted. The default grouping is as follows: Control/Address-0: CKE, CK, CKN, CS, RAS, CAS, WE, ODT, BA0, BA1, BA2, ALERT, ACTN, PARITY. Control/Address-1: BG0, BG1, A0, A1, A2, A3, A4, A5, A6, A7. Control/Address-2: A8, A9, A10, A11, A12, A13, A14, A15, A16.	Unchecked
	Control and Address Signals	This option box is displayed when the "Custom Control/Address Group" option is checked, and is used to select the Group and Pin where the control and address signals correspond to the PAD.	-
Data Pin Options	Bank Number	Select the Bank where DQ[5] is located.	R7
	Group Number	Select the Group where DQ[5] belongs to.	DQ[0-7]: G0 DQ[8-15]: G1
Rescal Mode Option	Rescal Mode[6]	Three optional calibration modes are supported. 1) Internal: Internal mode, this Bank IOB uses the calibration value of this Bank; 2) Adjacent Bank: Adjacent Bank mode, this Bank IOB uses the calibration value of the adjacent Bank, specifically set with the "Adjacent Bank Options" option; 3) User Config: user configuration mode, the user can configure the calibration value by himself. For detailed usage of GTP_RES_CAL_E1, please refer to: "UG050007_Titan2 Series GTP User Guide" [8].	Internal
	Adjacent Bank Options	Corresponding to "Adjacent Bank" calibration mode, it supports two options of UP and DN. This parameter is determined according to BANK selection.	-

Note: "-" indicates that the IP configuration interface has no default value for this parameter.

Attention:

• For the "Custom Control/Address Group" option, it is recommended to use a custom group because the PCB routing is different in actual use.

• When checking the "Custom Control/Address Group" option for custom pin configuration, make sure not to constrain the signals to the same pin. If there are pins bound to the same position, the UI interface will be marked in red to prompt the corresponding Pin Number.

---

[5] DQ[8-15] is displayed when the data bit width is greater than 8, DQ[16-23] is displayed when the data bit width is greater than 16, DQ[24-31] is displayed when the data bit width is greater than 24, DQ[32-39]] is displayed when the data bit width is greater than 32, DQ[40-47] is displayed when the data bit width is greater than 40, DQ[48-55] is displayed when the data bit width is greater than 48, DQ[56-63] is displayed when the data bit width is It is displayed at 56, and DQ[64-71] is displayed when the data bit width is greater than 64.

[6] IOB output impedance calibration option.

(4) Step 4: Summary

The Summary page is used to print the current configuration information, no configuration parameters are required, as shown in Figure 2-9.

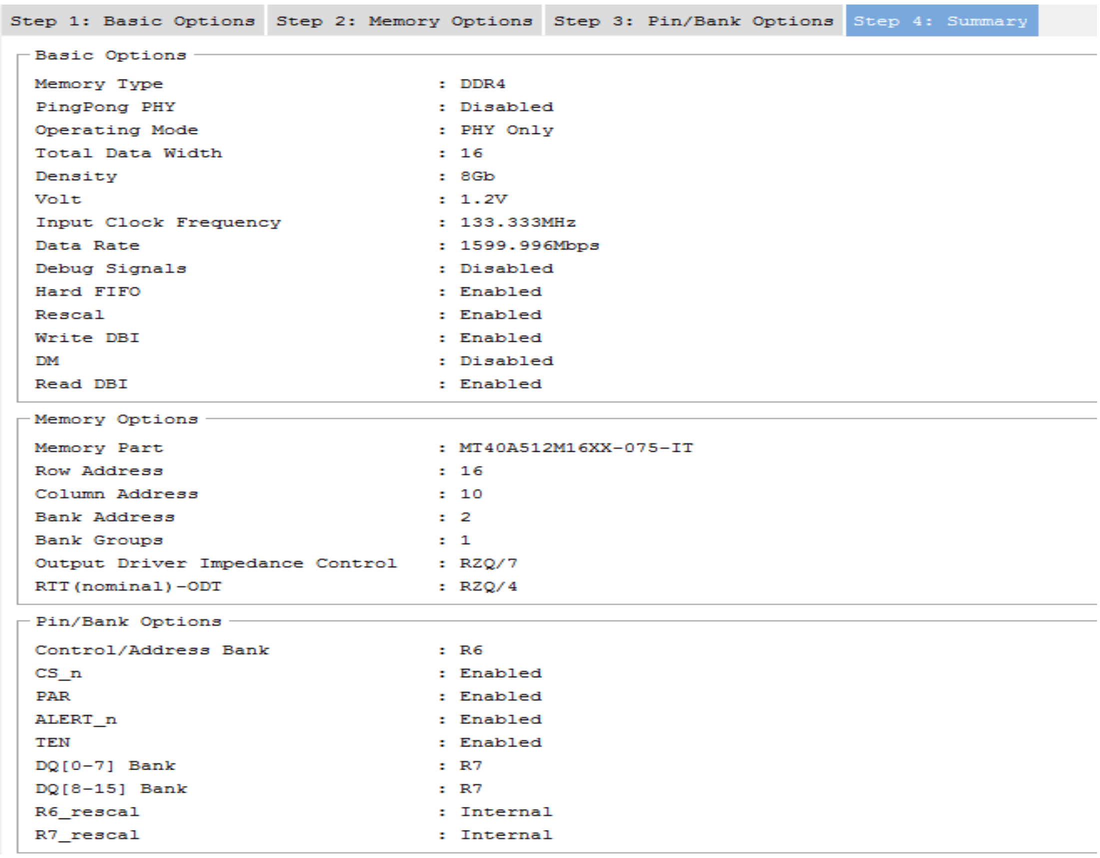


Figure 2-9 Summary page

3. Generate IP

After the parameter configuration is complete, click the <Generate> button on the upper left corner to generate the HMIC_S IP code corresponding to the user-specific settings. Figure 2-10 shows the report interface for generating IP information.

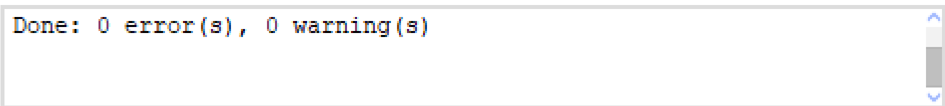


Figure 2-10 HMIC_S IP generate report interface

Attention:

The .pds file and .fdc file generated by the IP are for reference only, please change the pin assignment according to the actual pin connection when using it.

After the IP is successfully generated, the files shown in Table 2-5 will be output under the Project path specified in Figure 2-3.

Table 2-5 Output Files After IP Generation

Output File[7]	Description
$instname.v	The top-level .v file for the Controller + PHY of the generated IP.
$instname_ddrphy_top.v	The PHY top-level .v file for the generated IP.
$instname.idf	Configuration file for generated IP.
/rtl/*	RTL code file for generated IP.
/example_design/*	The Test Bench used by the Example Design of the generated IP and the corresponding Memory Simulation Model file.
/pnr/*	The corresponding project file .pds and pin assignment file .fdc of the generated IP Example Design.
/sim/*	Simulation directory for generated IP. sim.tcl is modelsim simulation script, makefile is vcs simulation script, sim_file_list.f is simulation file list.
/sim_lib/*	Directory of encrypted files required by IP.
/rev_1	Comprehensive report default output path. (This folder will only be generated if the synthesis tool is specified)
readme.txt	readme file, describing the structure of the generated directory after the IP is generated.

2.3.2 Constraint Configuration

For the specific configuration method of the constraint file, you can refer to the relevant help documents under the PDS installation path: User_Constraint_Editor_User_Guide[5], Physical_Constraint_Editor_User_Guide[6] , Route_Constraint_Editor_User_Guide[7].

2.3.3 Run Simulation

The simulation of HMIC_S IP is based on the Test Bench of Example Design. For details about Example Design, see "2.4 Example Design".

In the simulation application of HMIC_S IP, you can achieve acceleration and reduce simulation time by defining the macro IPS_DDR_SPEEDUP_SIM. The recommended definition method is as follows: Add the statement in the sim_file_list.f file under ../sim/modelsim or ../sim/vcs+define+IPS_DDR_SPEEDUP_SIM.

For more detailed information about the PDS simulation function and third-party simulation tools, you can refer to the relevant help documents under the PDS installation path: Pango_Design_Suite_User_Guide[2], Simulation_User_Guide[4].

[7] $instname is the instantiated name entered by the user; "*" is a wildcard, replacing files of the same type.

1. Modelsim simulation

Simulation running steps: Open cmd.exe, switch the current directory to /sim under the IP generation directory in the command line, run vsim, and open the ModelSim simulation software, as shown in Figure 2-11.

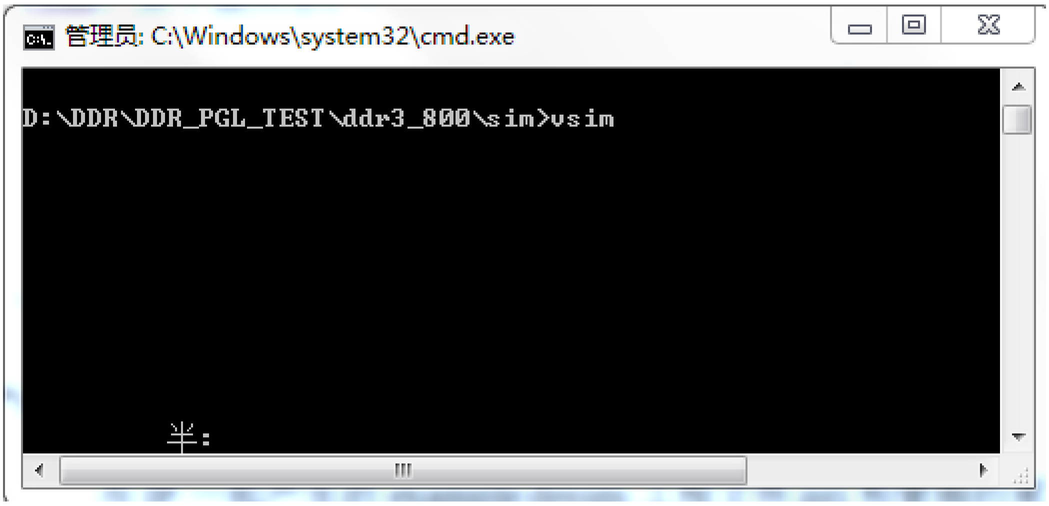


Figure 2-11 Open ModelSim command

Run the sim.tcl script in the ModelSim simulation software to perform simulation, as shown in Figure 2-12.

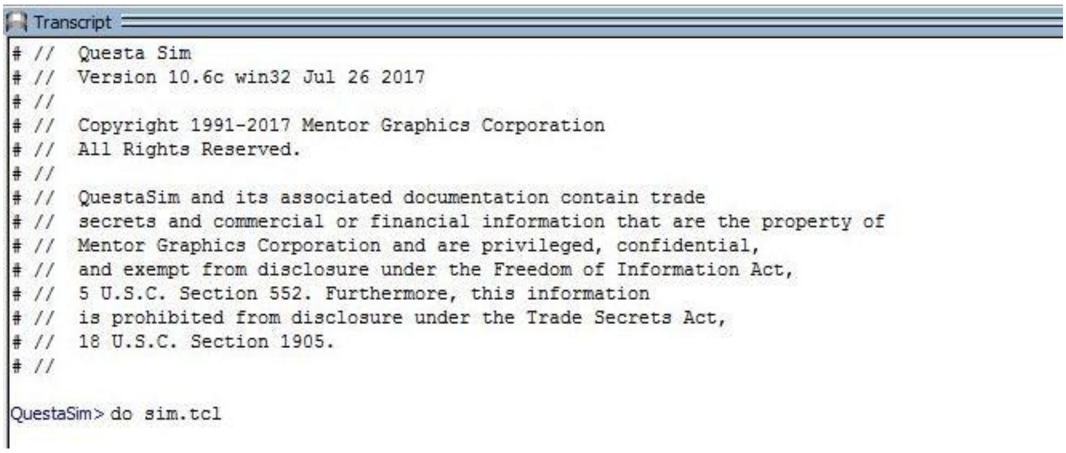


Figure 2-12 ModelSim executes the simulation script

2. VCS simulation

Enter the sim/vcs directory in Terminal, enter make and press Enter to start the simulation, as shown in Figure 2-13.



Figure 2-13 VCS executes the simulation script

2.3.4 Synthesis and Place and Route

For the specific usage of PDS synthesis tools and place and route tools, please refer to the help documents under the PDS installation path.

Attention:

The Example Design project file .pds and pin assignment file .fdc generated together with the IP are stored in the /pnr/ directory. When using it, it is necessary to modify the physical constraints according to the actual wiring of the board. For details, please refer to "2.8 Description and Precautions".

2.3.5 Resource Utilization

Table 2-6 HMIC_S IP Typical Resource Utilization Based on Applicable Devices

Device	Configuration Mode	Resource Utilization Typical				
		LUT	FF	GPLL	PPLL	USCM
PG2T390H	DDR4 x32 PHY Only	7435	7692	1	2	2
	DDR4 x64 PHY Only	11482	12633	1	3	2
	DDR4 PPPHY x32 PHY Only	11147	12205	1	3	2
	DDR3 x32 PHY Only	6974	7600	1	2	2
	DDR3 x64 PHY Only	8824	10259	1	3	2

2.4 Example Design

This section mainly introduces the Example Design solution based on HMIC_S IP (Controller + PHY mode). In this solution, the user logic acts as the AXI Master, and the HMIC_S IP acts as the AXI Slave. The user logic writes data through the Write channel of the AXI interface, and receives and compares the data through the Read channel of the AXI interface. If the data is wrong, the Error LED will be lit.

2.4.1 Design Block Diagram

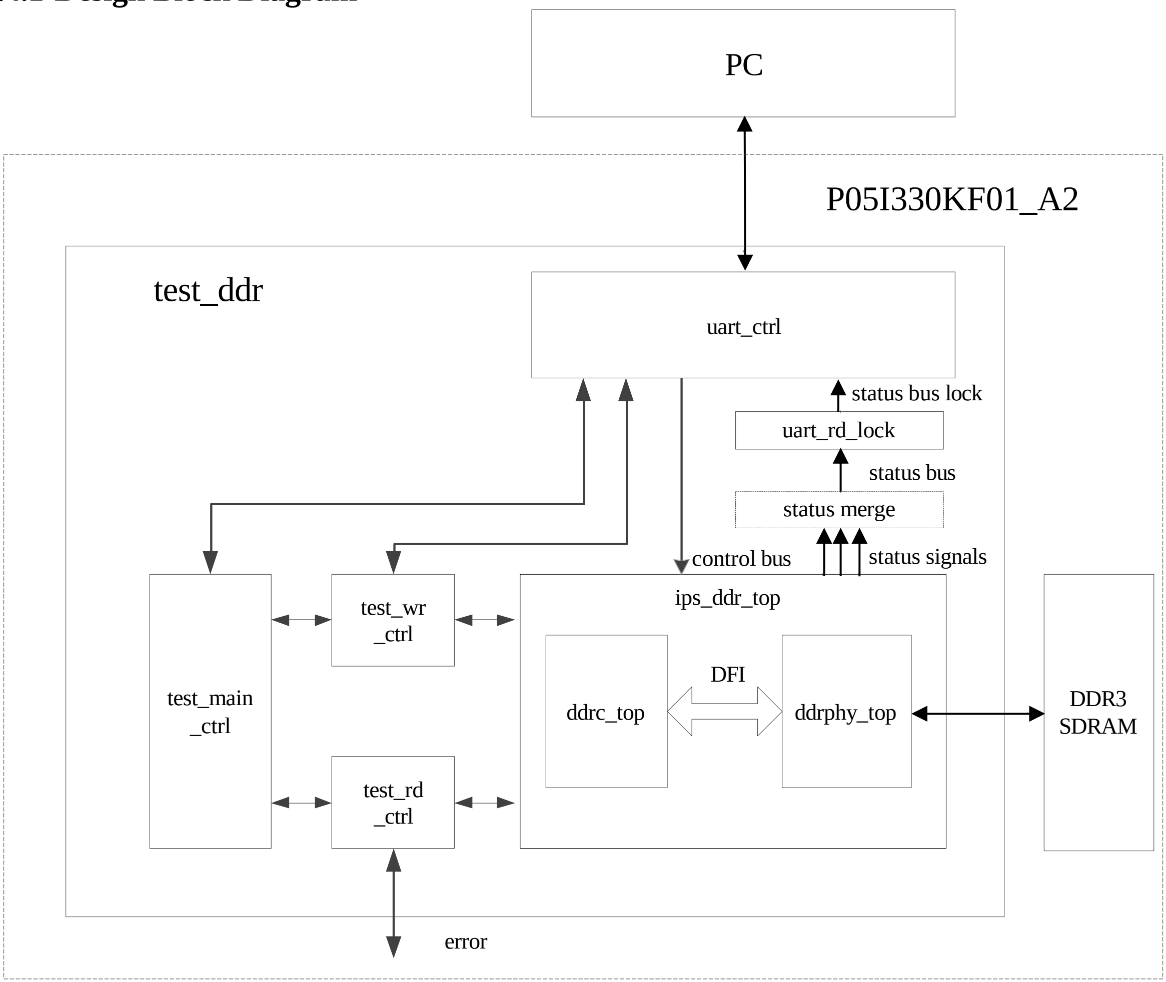


Figure 2-14 Example Design system block diagram

The system block diagram of Example Design is shown in Figure 2-14, where the test_main_ctrl module is the control module for AXI read and write instructions, the test_wr_ctrl module is the control module for AXI write instructions and write data, and the test_rd_ctrl module is the control module for AXI read instructions and read data, the uart_ctrl module is a serial port conversion module, which is convenient for controlling and reading the internal state during debugging.

2.4.2 Test method

In Example Design, the user logic reads and writes the HMIC_S IP and verifies the readback data. The detailed test process is shown in Figure 2-15.

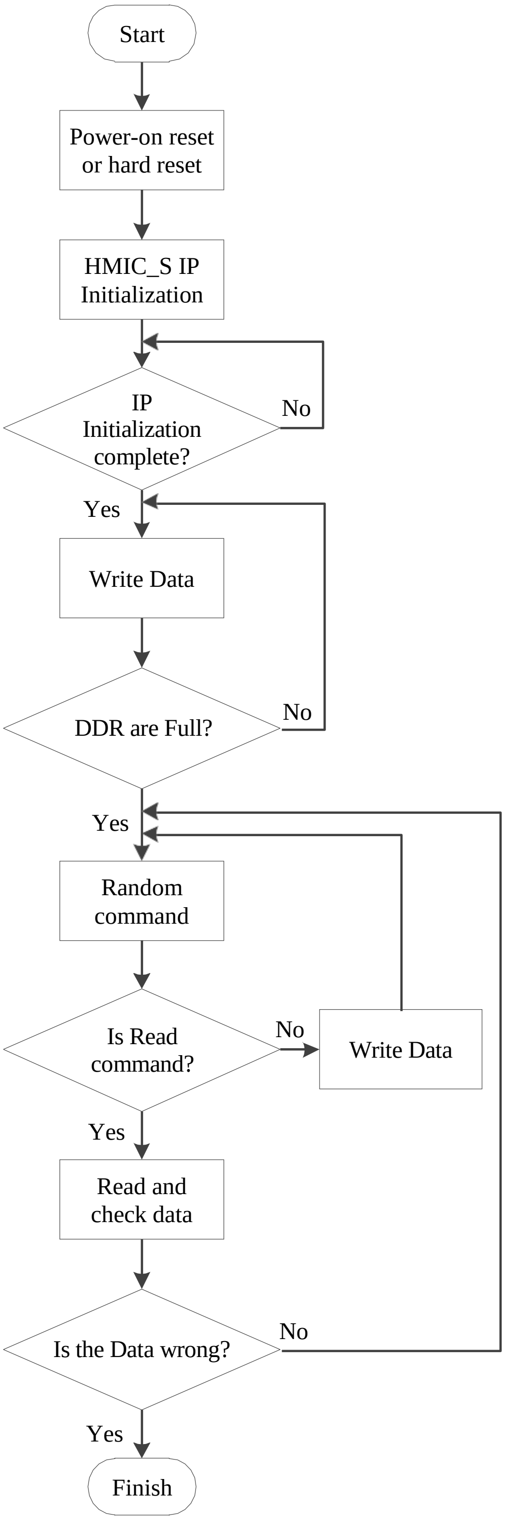


Figure 2-15 Example Design test flow chart

After the system is powered on or hard reset is started, HMIC_S IP starts to execute initialization. After the initialization is completed (ddrc_init_done is pulled up), the test_main_ctrl module controls the test_wr_ctrl module to generate write instructions and write data to initialize the data of DDR particles. After writing is full, test_main_ctrl starts to perform random Read and write, test_rd_ctrl checks the read-back data to determine whether the data is wrong

---

Attention:

The Example Design generated by IP cannot directly run the Flow on-board test. It is necessary to perform pin constraints according to the actual pin connection relationship of the single board, and then run the Flow on-board test.

2.5 IP Intarface description

This section introduces the HMIC_S IP related interface description and sequence description.

2.5.1 Controller Intarface description

1. Global Interface

Table 2-7 Global interface

Port	**I/O**	**Width**	**Valid Values**	**Description**
clk	I	1	-	External clock input.
rst_n	I	1	Low level	External reset input.
ddr_zqcs_req	I	1	High level	DDR ZQCS（ZQ calibration Short）command request.
ddr_zqcs_ack	O	1	High level	DDR ZQCS（ZQ calibration Short）command complete response.
phy_init_done	I	1	High level	The initialization completion flag of ddrphy: 1’b1: ddrphy initialization completed; 1’b0: ddrphy initialization is not complete.
ddr_init_done	O	1	High level	IP initialization complete flag: 1’b1: ddr IP initialization completed; 1’b0: The ddr IP initialization is not completed, and the external operation on the ddr IP is invalid.

Note: "-" indicates that there is no such parameter.

2. AXI4 Interface

(1) Write Address Channel

Table 2-8 Write address channel

Port	I/O	Width	Valid Values	Description
axi_awaddr	I	AXI_ADDR_WIDTH	-	AXI write address.
axi_awid	I	8	-	AXI write address ID.
axi_awsize	I	3	-	AXI write burst size.
axi_awburst	I	2	-	AXI write burst type.
axi_awlen	I	8	-	AXI write burst length.
axi_awready	O	1	High level	AXI write address ready.
axi_awvalid	I	1	High level	AXI write address valid.

Note: "-" indicates that there is no such parameter.

(2) Read Address Channel

Table 2-9 Read address channel

Port	I/O	Width	Valid Values	Description
axi_araddr	I	AXI_ADDR_WIDTH	-	AXI read address.
axi_arid	I	8	-	AXI read address ID.
axi_arsize	I	3	-	AXI read burst size.
axi_arburst	I	2	-	AXI read burst type.
axi_arlen	I	8	-	AXI read burst length.
axi_arready	O	1	High level	AXI read address ready.
axi_arvalid	I	1	High level	AXI read address valid.

Note: "-" indicates that there is no such parameter.

(3) Write Data Channel

Table 2-10 Write data channel

Port	I/O	Width	Valid Values	Description
axi_wdata	I	AXI_DATA_WIDTH*8	-	AXI write data.
axi_wstrb	I	AXI_DATA_WIDTH*8/8	High level	AXI write data strobes.
axi_wvalid	I	1		AXI write data valid.
axi_wready	O	1	High level	AXI write data ready.
axi_bready	I	1	High level	AXI writes the reply ready.
axi_bid	O	8	-	AXI write data ID.
axi_bresp	O	2	-	AXI writes the reply.
axi_bvalid	O	1		AXI write reply valid.

Port	I/O	Width	Valid Values	Description
axi_wlast	I	1	High level	AXI write data last.

Note: "-" indicates that there is no such parameter.

(4) Read Data Channel

Table 2-11 Read data channel

Port	I/O	Width	Valid Values	Description
axi_rid	O	8	-	AXI read data ID.
axi_rlast	O	1	High level	AXI read data last signal.
axi_rvalid	O	1	High level	AXI read data valid.
axi_rdata	O	AXI_DATA_WIDTH*8	-	AXI read data.
axi_rready	I	1		AXI read data ready. Note: axi_rready needs to be pulled high when axi_rvalid is valid.
axi_rresp	O	2		AXI read reply.

Note: "-" indicates that there is no such parameter.

3. Config Interface

Table 2-12 Config interface

Port	I/O	Width	Valid Values	Description
apb_clk	I	1	High level	APB clock.
apb_rst_n	I	1	Low level	APB reset.
apb_sel	I	1	High level	APB Select.
apb_enable	I	1	High level	APB port enable.
apb_addr	I	8	-	APB Address bus.
apb_write	I	1	High level	APB read and write direction, high level write, low level read.
apb_ready	O	1	High level	APB port Ready.
apb_wdata	I	16	-	APB write data.
apb_rdata	O	16	-	APB read data.

Note: "-" indicates that there is no such parameter.

4. DFI Interface

Table 2-13 DFI interface

Port	I/O	Width	Valid Values	Description
dfi_address	O	4*ROW_ADDR_WIDTH	-	DFI address bus.
dfi_bank	O	4*BADDR_WIDTH	-	DFI bank. Address bus.

Port	I/O	Width	Valid Values	Description
dfi_cs_n	O	4	-	DFI chip select.
dfi_ras_n	O	4	-	DFI row address strobe bus.
dfi_cas_n	O	4	-	DFI column address strobe.
dfi_we_n	O	4	-	DFI write enable signal.
dfi_cke	O	4	-	DFI clock enable.
dfi_odt	O	4	-	DFI on-die termination control bus.
dfi_rddata_valid	I	1	High level	The read data valid of the dfi interface.
dfi_rddata	I	DQ_WIDTH*8	-	Data for the dfi interface.
dfi_wrdata_en	O	4	High level	DFI write enable signal.
dfi_wrdata	O	DQ_WIDTH*8	-	DFI write data.
dfi_wrdata_mask	O	DQ_WIDTH*8/8	High level	DFI Write data byte mask.
dfi_phyupd_req	I	1	-	The PHY requests an update.
dfi_phyupd_ack	O	1	-	Feedback signal for dfi_phy_req, allowing The PHY is updated.

Note: "-" indicates that there is no such parameter.

2.5.2 Controller Interface Timing Description

The standard AXI4 protocol used by the AXI4 interface. The Config interface uses the APB protocol.

1. AXI4 Inferface

The AXI4 interface includes 4 channels, which are the write address channel, read address channel, write data channel, and read data channel, and each channel is independent of each other. Generally speaking, you need to send the address first, and then send and receive data. Users can judge whether the returned data is the requested value through different id numbers.

There are two correspondences between the AXI4 interface address and the Memory address, corresponding to the "Memory Address Mapping Selection" option area of the configuration interface. When "ROW+BANK+COLUMN" is selected, the corresponding relationship between AXI4 interface address and Memory address is shown in Figure 2-16; when "BANK+ROW+COLUMN" is selected, the corresponding relationship between AXI4 interface address and Memory address is shown in Figure 2-17.

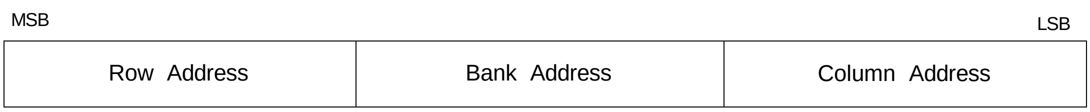


Figure 2-16 AXI4 interface address mapping 1

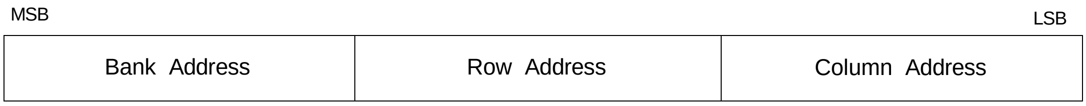


Figure 2-17 AXI4 interface address mapping 2

(1) Write address channel timing

The signals included in the write address channel: axi_awready, axi_awvalid, axi_awaddr, axi_awsize, axi_awburst, axi_awid, axi_awlen. Typical timing is shown in Figure 2-18.

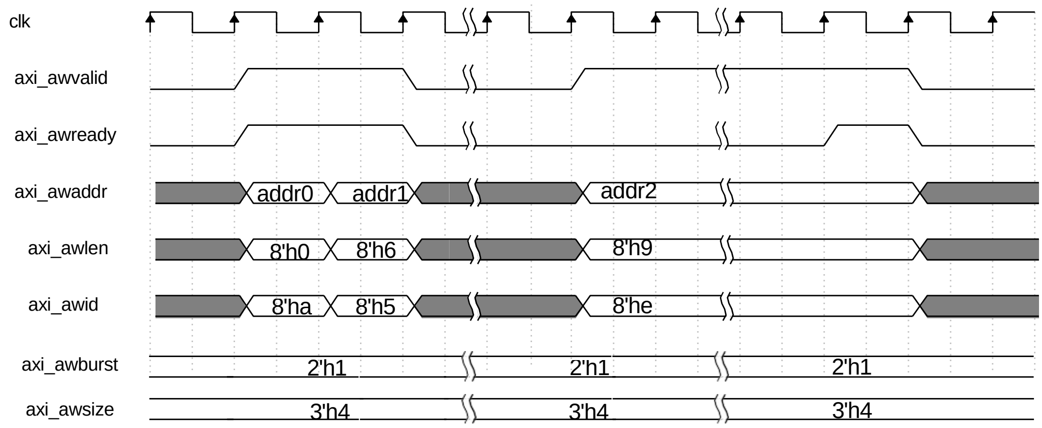


Figure 2-18 Typical Timing of Write Address

- One handshake completion condition: axi_awready and axi_awvalid are valid at the same time.
- The packet length is controlled by axi_awlen, and the packet length is the value of axi_awlen plus 1.
- Handshaking process: starting from the rising edge of the axi_awvalid valid clock, axi_awaddr, axi_awid, and axi_awlen need to remain unchanged until the handshake is completed and released.

(2) Read address channel timing

The signals contained in the read address channel: axi_arready, axi_arvalid, axi_araddr, axi_arburst, axi_arsize, axi_arid, axi_arlen. Typical timing is shown in Figure 2-19.

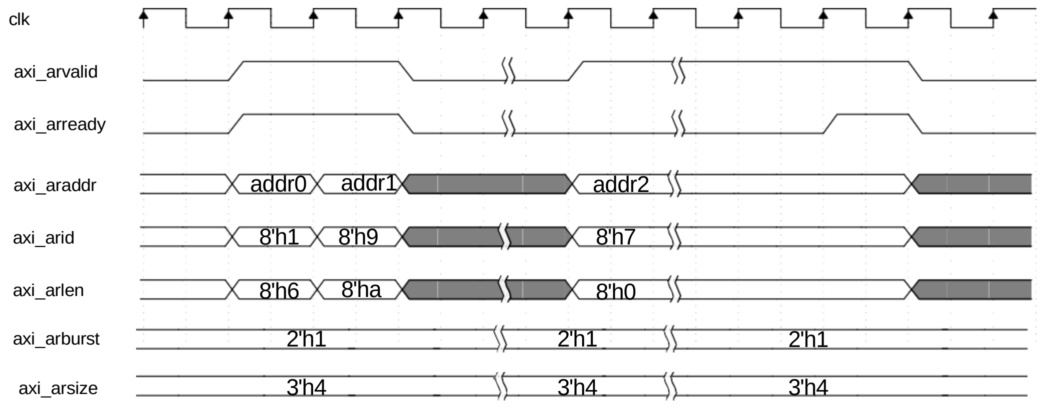


Figure 2-19 Typical Timing of Read Address

- One handshake completion condition: axi_arready and axi_arvalid are valid at the same time.
- The packet length is controlled by axi_arlen, and the packet length is the value of axi_arlen plus 1.
- Handshake process: starting from the rising edge of the axi_arvalid valid clock, axi_araddr, axi_arid, and axi_arlen need to remain unchanged until the handshake is completed and released.

(3) Write data channel timing

The write data channel contains signals: axi_wready, axi_wvalid, axi_bready, axi_bid, axi_bresp, axi_bvalid, axi_wlast, axi_wdata and axi_wstrb. Typical timing is shown in Figure 2-20.

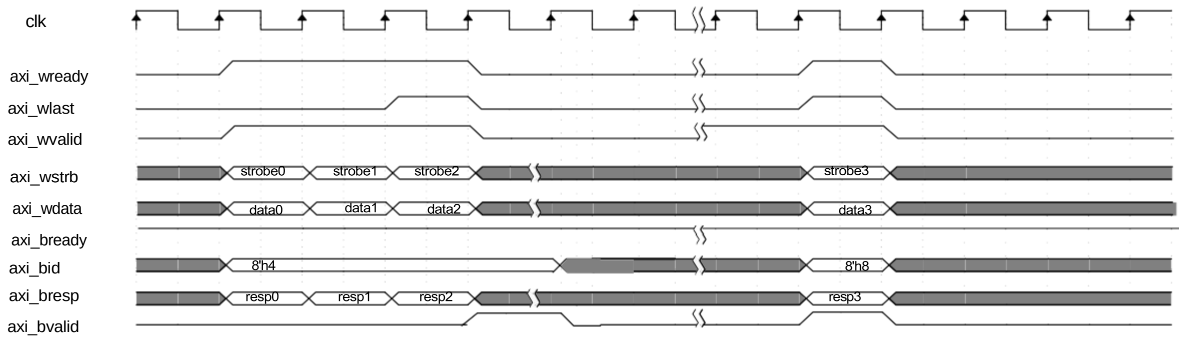


Figure 2-20 Typical timing sequence for writing data

- Transmission valid indication: axi_wready is valid.
- Indication of the end of a transmission: axi_wlast is valid.
- During transmission: axi_wready, axi_bid and axi_wlast are received synchronously, axi_wdata and axi_wstrb are sent synchronously.

(4) Read Data Channel Timing

Read data channel Contains signals: axi_rdata, axi_rready, axi_rresp, axi_rid, axi_rlast and axi_rvalid. Typical timing is shown in Figure 2-21.

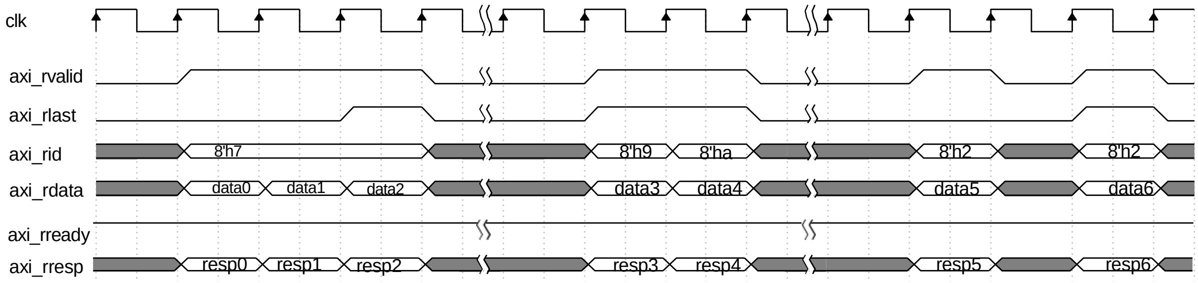


Figure 2-21 Typical timing sequence for reading data

- Transmission valid indication: axi_rvalid is valid.
- Indication of the end of a transmission: axi_rlast is valid.
- axi_rready needs to be pulled high when axi_rvalid is valid.

2. Config Interface

The Config interface adopts the standard APB protocol. By configuring the corresponding registers, DDR3/DDR4 SDRAM can be switched between Power Down, Self-Refresh, MRS and Normal states. By reading the corresponding register, the current state of DDR3/DDR4 can be queried.

The APB interface is a half-duplex communication, the read and write data lines are independent, and the control line and address line are multiplexed. Each handshake needs to consume at least 2 cycles of apb_clk.

The signals included in the APB interface: apb_enable, apb_ready, apb_rdata, apb_sel, apb_write, apb_wdata, apb_clk.

A handshake completion condition: apb_enable and apb_ready are valid at the same time.

(1) APB interface write timing

The typical write sequence of APB interface is shown in Figure 2-22.

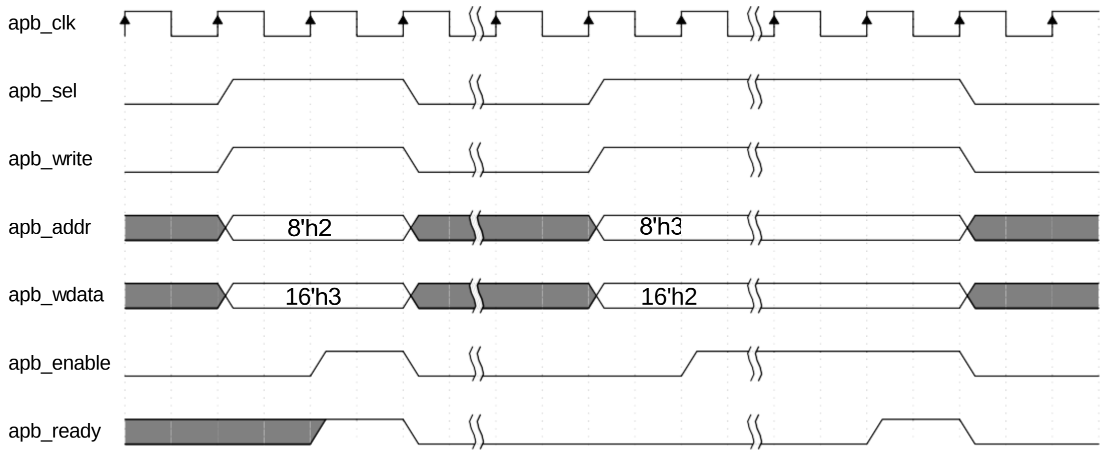


Figure 2-22 Typical write timing of APB interface

- The first clock cycle: apb_sel, apb_write are pulled high, apb_addr, apb_wdata are given initial values, they need to remain stable until they are released after the handshake is completed.
- The second clock cycle: apb_enable is pulled high until the handshake is completed and released.

(2) APB interface read timing

The typical read timing of APB interface is shown in Figure 2-23.

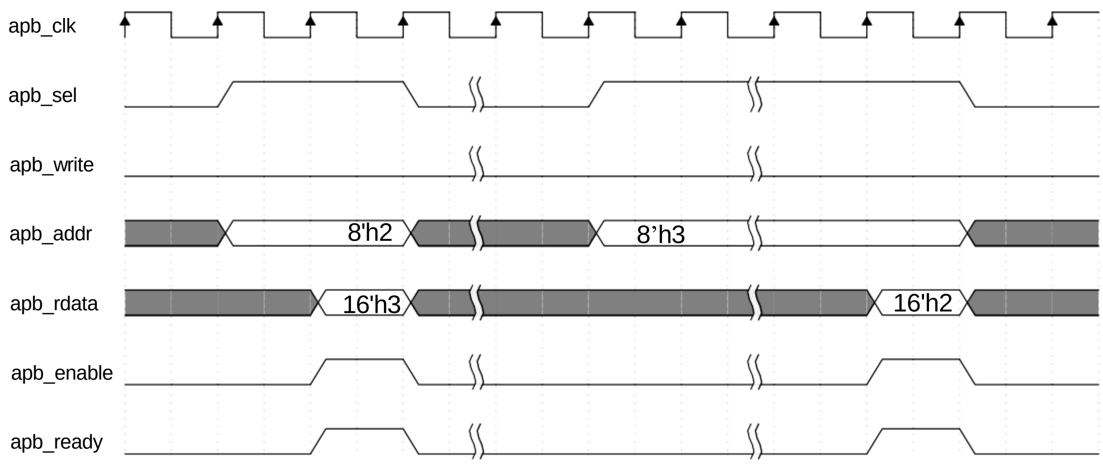


Figure 2-23 Typical read timing of APB interface

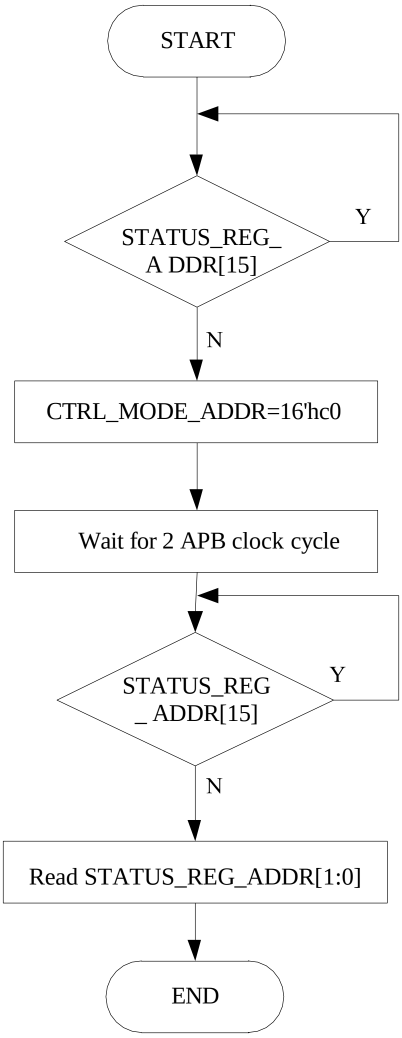
- The first clock cycle: apb_sel is pulled high, apb_write is pulled low, apb_addr is given the initial value, and they need to remain stable until the handshake is completed and released.
- The second clock cycle: apb_enable is pulled high until the handshake is completed and released.
- Valid data: apb_rdata is only valid during handshake.

(3) DDR3/DDR4 State switching and query methods

In order to avoid the user's state switching request from interfering with the normal DDR3/DDR4 timing, the user needs to follow certain operating rules when performing state switching (query the STATUS_REG_ADDR register to determine the current state of DDR3/DDR4 and whether the request has been responded; control the CTRL_MODE_DATA register to realize DDR3/DDR4 state switching).

- When sending a new state switching request, you must first check whether the current state switching request has been responded, otherwise an error will result.
- The state switch request and the state switch trigger enable need to be sent synchronously, that is, bit 0 and bit[15:14] of CTRL_MODE_DATA need to be configured at the same time.
- When configuring the Mode Registers inside DDR, you need to configure the MODE_REG_0_ADDR, MODE_REG_1_ADDR, MODE_REG_2_ADDR, MODE_REG_3_ADDR, MODE_REG_4_ADDR, MODE_REG_5_ADDR, MODE_REG_6_ADDR registers first, and then configure CTRL_MODE_DATA register.
- After sending the state switching request, wait for 2 apb_clks, and then check the relevant state register to see if it has responded.

The state request switching process is shown in Figure 2-24, taking controlling DDR3/DDR4 to enter the Power Down state as an example.



Step1: Query whether the current state is responded

Step2: Send new status request

Step3: Wait for 2 APB clock cycles

Step4: Query whether the new state is responded

Step5: The new request has responded, read the new status

Figure 2-24 DDR3/DDR4 state request switching flow chart

2.5.3 PHY interface description

1. Clock and reset interface

Table 2-14 Clock and Reset Interface

Port	I/O	Width	Valid Values	Description
ref_clk	I	1	-	External Reference Clock Input.
ddr_rstn	I	1	-	External reset input.
pll_lock	O	1	High level	PLL lock indication, high level means locked.
ddrphy_cpd_lock	O	1	High level	cpd lock indication, high level means locked.
ddrphy_sysclk	O	1	-	The system clock of the PHY is output to the working clock of the controller.

Note: "-" indicates that there is no such parameter.

2. DFI Interface

Table 2-15 DFI interface

Port	I/O	Width	Valid Values	Description
dfi_address	I	4*ROW_ADDR_WIDTH	-	DFI Address Bus.
dfi_bank	I	4*BADDR_WIDTH	-	DFI bank. Address Bus.
dfi_reset_n	I	1	-	DFI chip reset.
dfi_cs_n	I	4	-	DFI chip select.
dfi_bg	I	4*MEM_BANKGROUP_WIDTH	-	DFI Bank Group (DDR4 only).
dfi_act_n	I	4	-	DFI Activation Command (DDR4 only).
dfi_ras_n	I	4	-	DFI row address strobe bus. (DDR3 only).
dfi_cas_n	I	4	-	DFI column address strobe. (DDR3 only).
dfi_we_n	I	4	-	DFI write enable signal. (DDR3 only).
dfi_cke	I	4	-	DFI clock enable.
dfi_odt	I	4	-	DFI on-die termination control bus.
dfi_rddata_valid	O	1	High level	The read data of the dfi interface is valid.
dfi_rddata	O	DQ_WIDTH*8	-	Data for the dfi interface.
dfi_wrdata_en	I	4	High level	DFI write enable signal.
dfi_wrdata	I	DQ_WIDTH*8	-	DFI write data.
dfi_wrdata_mask	I	DQ_WIDTH*8/8	High level	DFI Write data byte mask.
dfi_init_complete	O	1	High level	The PHY has completed the training-related operations and is in the normal state.
dfi_error	O	1	High level	PHY training error indication.
dfi_phyupd_req	O	1	-	The PHY requests an update.
dfi_phyupd_ack	I	1	-	Feedback signal of dfi_phy_req, allowing PHY to update.

Note: "-" indicates that there is no such parameter.

3. Memory Interface

Description:

The Memory interface in this IP is subject to the agreement. If the granule selected by the user includes an interface that is not included in the protocol, please refer to the corresponding granule Datasheet to add it yourself and handle it properly.

Table 2-16 Memory interface

Port	I/O	Width	Valid Values	Description
mem_a	O	ROW_ADDR_WIDTH	-	DDR row and column address bus.
mem_ba	O	BANK_ADDR_WIDTH	-	DDR Bank address.
mem_ck	O	1	-	DDR input system clock.
mem_ck_n	O	1	-	DDR input system clock.
mem_cke	O	1	High level	DDR input system clock is valid.
mem_dm	O	DM_WIDTH	High level	DDR input data Mask.
mem_odt	O	1	-	DDR ODT.
mem_cs_n	O	1	Low level	Chip select for DDR.
mem_ras_n	O	1	Low level	Row address enable (DDR3 only).
mem_cas_n	O	1	Low level	Column address enable (DDR3 only).
mem_we_n	O	1	Low level	DDR write enable signal (DDR3 only).
mem_rst_n	O	1	Low level	DDR reset.
mem_dq	I/O	DQ_WIDTH	-	DDR data.
mem_dqs	I/O	DQS_WIDTH	-	DDR data-associated clock.
mem_dqs_n	I/O	DQS_WIDTH	-	DDR data-associated clock.
mem_bg	O	MEM_BANKGROUP_WIDTH	-	DDR bank group (DDR4 only).
mem_act_n	O	1	-	DDR Activation Command (DDR4 only).
mem_parity	O	1	-	DDR Command and Address Parity (DDR4 only).
mem_alert_n	I	MEM_ALERT_ADDRESS	-	Memory error flag (DDR4 only).

Note: "-" indicates that there is no such parameter.

4. Debug Interface

Description:

For the specific usage of the debugging interface, please refer to "UG050010_Titan2 Series HMIC_S IP Application Guide" [12].

Table 2-17 Debug Interface

Port	I/O	Width	Description
Manually debug input signals			
debug_gate_start	I	1	Reset sequence control, valid on rising edge.
debug_cpd_start	I	1	Reset sequence control, valid on rising edge.
debug_ddrphy_rst_n	I	1	Reset sequence control, active low.
debug_gpll_scan_rst	I	1	Reset sequence control, active high.
debug_dll_update_en	I	1	dll update enable signal
init_wrlvl_step	I	8*DQS_WI DTH	The initial step of dqs when doing Write leveling. There are DQS_WIDTH groups, each group is 8bit, and each group can be configured independently.
init_slip_step	I	4*DQS_WI DTH	The initial value of dqs gate coarse tuning position. There are a total of DQS_WIDTH groups, each group is 4bit, and each group can be configured independently.
init_read_clk_ctrl	I	3*DQS_WI DTH	The initial value of dqs gate fine-tuning position. There are a total of DQS_WIDTH groups, each group is 3bit, and each group can be configured independently.
ck_dly_set_bin	I	8	The CK signal output delay adjustment port of the memory interface. The port configuration value is multiplied by 5ps, which is the additional delay value of CK, which is calculated based on the demo board routing delay. The additional delay value of CK is preferably in the range of 50~150ps.
dqs_even_adj	I	8	Read direction dqs rising edge delay adjusts the port. bit7 is the sign bit, 1 means decrease delay, 0 means increase delay. The bit6-0 configuration value is multiplied by 5ps, which is the additional delay value added on the rising edge of dqs.
dqs_odd_adj	I	8	Read direction dqs falling edge delay adjusts the port. bit7 is the sign bit, 1 means decrease delay, 0 means increase delay. Multiplying the bit6-0 configuration value by 5ps is the additional delay value added on the falling edge of dqs.

Port	I/O	Width	Description
force_read_clk_ctrl	I	1	The dqs gate position is fixed and enabled, and the high level is effective. 0: The dqs gate position changes during training. 1: The position of the dqs gate remains unchanged and is always the initial value.
rescal_pcode1_bank0~2	I	5	IOB output impedance calibration value input, pcode, the first set of values, bank0~3 correspond to bank R5, R6 or R7.
rescal_ncode1_bank0~2	I	5	IOB output impedance calibration value input, ncode, the first set of values, bank0~3 correspond to bank R5, R6 or R7.
rescal_pcode2_bank0~2	I	5	IOB output impedance calibration value input, pcode, the second set of values, bank0~3 correspond to bank R5, R6 or R7.
rescal_ncode2_bank0~2	I	5	IOB output impedance calibration value input, ncode, the second set of values, bank0~3 correspond to bank R5, R6 or R7.
rescal_rst	I	1	rescal training reset enable, active high. Reset once to start training once.
gate_update_en	I	1	dqs gate update enable, active high.
dbg_manu_gate_upd_en	I	1	dqs gate update manual control enable, active high.
dbg_manu_gate_upd_sta	I	1	dqs gate update Manual control starts, the rising edge is valid.
upd_act_ba	I	MEM_BANKADDR_WIDTH	Active bank during dqs gate update.
upd_act_bg	I	MEM_BANKGROUP_WIDTH	Active group during dqs gate update.
gate_dqs_sample_sel	I	1	Gate result judgment signal selection during dqs gate update: 0: select dqs_gate_sample; 1: select ddrphy_dgts;
left_err_sure_num	I	5	dqs gate update 16 samples left to adjust the decision threshold. When the number of errors is greater than or equal to the threshold, it is judged as error, otherwise it is judged as correct.
right_err_sure_num	I	5	dqs gate update 16 samples right to adjust the decision threshold. When the number of errors is greater than or equal to the threshold, it is judged as error, otherwise it is judged as correct.
debug_cpd_offset_adj	I	1	cpd phase adjustment enable, rising edge is valid.
debug_cpd_offset_dir	I	1	cpd phase adjustment direction. 0: GPLL clock phase shifted right. 1: Shift the GPLL clock phase to the left.
force_wrlvl	I	1	write leveling manual configuration enable, active high.
force_wrlvl_step	I	8*MEM_DQS_WIDTH	write leveling configuration value. Force_wrlvl is valid when high level.

Port	I/O	Width	Description
force_dqsgate	I	1	dqs gate manual configuration enable, active high.
force_read_clk_ctrl	I	3*MEM_DQS_WIDTH	Configuration value for dqs gate fine-tuning location. Force_dqsgate is valid when high.
force_slip_step	I	4*MEM_DQS_WIDTH	Configuration value for dqs gate coarse tuning position. Force_dqsgate is valid when high.
force_rdvref	I	1	Read vref Manual configuration enable, active high.
force_rdvref_range	I	1	Read vref to manually configure range settings. 0: range1; 1: range2; force_rdvref is valid when high level.
force_rdvref_value	I	6*MEM_DQS_WIDTH	Read vref to manually configure the value. Force_rdvref is valid when high level.
force_rdcal	I	1	Read calibration manual configuration enable, active high.
force_rd_dq_dly	I	64*MEM_DQS_WIDTH	Read dq calibration manual configuration value. Force_rdcal is valid when high level.
force_dqs_even_dly	I	8*MEM_DQS_WIDTH	Read dqs Calibration rising edge manual configuration value. Force_rdcal is valid when high level.
force_dqs_odd_dly	I	8*MEM_DQS_WIDTH	Read dqs Calibration falling edge manual configuration value. Force_rdcal is valid when high level.
force_rd_dbi_dly	I	8*MEM_DQS_WIDTH	Read dbi calibration manual configuration value. Force_rdcal is valid when high level.
force_wrvref	I	1	Write vref calibration manual configuration enable, active high.
force_wrvref_range	I	1	Write vref to manually configure range settings. 0: range1; 1: range2; force_wrvref is valid when high level.
force_wrvref_value	I	6	Write vref to manually configure the value. Force_wrvref is valid when high level.
force_wrcal	I	1	Write calibration manual configuration enable, active high.
force_wr_dq_dly	I	64*MEM_DQS_WIDTH	Write dq to calibrate manual configuration values. Force_wrcal is valid when high level.
force_wrdm_dbi_dly	I	8*MEM_DQS_WIDTH	Write data mask(dbi) to manually configure the value. Force_wrcal is valid when high level.
wr_win_scan_en	I	1	Write scan window enable, active high.
wr_win_scan_group_en	I	MEM_DQS_WIDTH	Write sweep window group enable, active high. wr_win_scan_en is valid when high level.

Port	I/O	Width	Description
rd_win_scan_en	I	1	Read sweep window enable, active high.
rd_win_scan_group_en	I	MEM_DQS_WIDTH	Read sweep window group enable, active high. rd_win_scan_en is valid when high level.
align_error_clr	I	1	Read direction group alignment error latch clear signal, the rising edge clears align_error_lck.
Common Debug Output Signals			
Debug data			
debug_data[8]	O	66*DQS_WIDTH	For each group of DDRPHY's Debug data, 8bit DQ share one DDRPHY.
debug_calib_ctrl[9]	O	36	Debug data in Tainning state.
debug_slice_status[10]	O	17*MEM_DQS_WIDTH	Training status.
debug_slice_state[11]	O	22*MEM_DQS_WIDTH	Training status.
Other			
debug_dll_upd_state	O	2	dll update controls the state.
debug_gpll_dps_phase	O	9	gpll current phase.
debug_rst_state	O	4	Reset sequence state.
debug_eye_dq_win	O	72*MEM_DQS_WIDTH	Eye calibration dq window.
debug_dq_dly_bin	O	64*MEM_DQS_WIDTH	dq delay binary code.
debug_eye_dm_win	O	9*MEM_D QS_WIDTH-1	Eye calibration dm window.
debug_dm_dly_bin	O	8*MEM_DQS_WIDTH	dm delay binary code.
debug_rd_dq_dly_bin	O	64*MEM_DQS_WIDTH	Read direction dq delay binary code.
debug_rd_dbi_dly_bin	O	8*MEM_DQS_WIDTH	Read direction dbi delay binary code.
debug_rdvref_margin_win	O	7*MEM_DQS_WIDTH	Read vref Calibration window.
debug_rdvref_left_edge	O	6*MEM_DQS_WIDTH	Read vref to calibrate the left margin.
debug_rdvref_mode	O	6*MEM_D	Read vref calibration value.

[8] debug_data See "2.5.5 1 debug_data description" for the meaning description of each field.
[9] debug_calib_ctrl See "2.5.5 2 debug_calib_ctrl description" for the meaning description of each field.
[10] debug_slice_status See "2.5.5 3 debug_slice_status description" for the meaning description of each field.
[11] debug_slice_state See "2.5.5 4 debug_slice_state description" for the meaning description of each field.

Port	I/O	Width	Description
		QS_WIDTH	
pcode_out	O	30	IOB output impedance calibration value output: bit[29:25]: pcode odd of bank2; bit[24:20]: pcode even of bank2; bit[19:15]: pcode odd of bank1; bit[14:10]: pcode odd of bank1 pcode even of bank1; bit[9:5]: pcode odd of bank0; bit[4:0]: pcode even of bank0.
ncode_out	O	30	IOB output impedance calibration value output: bit[29:25]: ncode odd of bank2; bit[24:20]: ncode even of bank2; bit[19:15]: ncode odd of bank1; bit[14:10]: ncode odd of bank1 ncode even of bank1; bit[9:5]: ncode odd of bank0; bit[4:0]: ncode even of bank0.
ddrphy_update_done	O	1	dqs gate update completion indication: 1: update completed; 0: update not completed;
dqs_drift	O	2*MEM_DQS_WIDTH	dqs Drift indicator signal.
dbg_dram_vref_left_edge_phy0	O	6	dram vref calibrates the left boundary value.
dbg_dram_vref_margin_win_phy0	O	7	dram vref Calibration window value.
dbg_dram_vref_range	O	1	dram vref calibration range.
dbg_dram_vref_value_phy0	O	6	dram vref calibration value.
dbg_update_done	O	1	dqs gate update completion indication signal, active high.
left_sample_error	O	16*MEM_DQS_WIDTH	dqs gate update 16 left-adjusted sampling results.
right_sample_error	O	16*MEM_DQS_WIDTH	dqs gate update 16 times to right adjust the sampling result.
left_sample_err_sure	O	MEM_DQS_WIDTH	dqs gate update 16 samples to left adjust the decision result.
right_sample_err_sure	O	MEM_DQS_WIDTH	dqs gate update 16 samples to right adjust the decision result.
left_adj_cnt	O	8*MEM_DQS_WIDTH	dqs gate update adjusts count left after multiple verdicts.
right_adj_cnt	O	8*MEM_DQS_WIDTH	dqs gate update adjusts the count to the right after multiple judgments.
no_adj_cnt	O	8*MEM_DQS_WIDTH	dqs gate update does not adjust the count after multiple judgments.

Port	I/O	Width	Description
debug_cpd_offset	O	10	cpd Phase adjustment value.
debug_cpd_state	O	4	cpd phasing status.
align_error	O	1	Read direction group data alignment error signal: 0: group data alignment is correct; 1: group data alignment error; This signal is a dynamic signal.
align_error_lck	O	1	Read direction group data alignment error latch signal: 0: group data alignment is correct; 1: group data alignment error; This signal is the latch signal of align_error, A rising edge on align_error_clr clears this signal.

Attention

After the Training is completed, the user can treat all Debug ports as asynchronous signals, and optimize the timing by setting False Path if necessary. Among them, the signal values of coarse_slip_step and read_clk_ctrl can only be read when dfi_phyupd_ack is pulled low, otherwise their signal values may change during gate update, resulting in inaccurate data reading.

2.5.4 PHY Interface Timing Description

1. DFI Interface Specification

When the user directly accesses the PHY Layer, it needs to comply with the DFI-like interface specification defined in this article. Through this interface, users can perform the following operations.

- Access (read/write) DDR SDRAM;
- Put DDR3 into Power Down or Self-Refresh state;
- Dynamically configure the value of DDR SDRAM internal registers;
- Read the status of DDR SDRAM.

For the differences between the DFI interface used in this design and the standard DFI 3.1 Specification[11], see Table 2-18 and Table 2-19.

Table 2-18 Differences between the DFI interface of this design and the standard DFI

Interface	Difference
ControlInterface	No requirement for tctrl_delay and tcmd_lat.
Write DataInterface	Compatible with standard DFI except the dfi_wrdata_cs_n signal.
Read DataInterface	Only the dfi_rddata and dfi_rddata_valid signals are kept.
UpdateInterface	Adopt a custom interface.
StatusInterface	Only dfi_init_complete is left.
TrainingInterface	NA.
Low PowerControlInterface	NA.
ErrorInterface	Compatible.

Table 2-19 Detailed difference comparison between DFI interface and standard DFI

Interface Signal Class	Standard DFI	Simplified DFI	Source	Description
Control signals	dfi_act_n	dfi_act_n	MC	DFI Activation Command (DDR4 only)
	dfi_address	dfi_address	MC	DFI address bus
	dfi_bank	dfi_bank	MC	DFI bank address bus
	dfi_bg	dfi_bg	MC	DFI Bank Group (DDR4 only)
	dfi_cas_n	dfi_cas_n	MC	DFI column address strobe (DDR3 only)
	dfi_cid	NA	MC	DFI Chip ID
	dfi_cke	dfi_cke	MC	DFI clock enable
	dfi_cs_n	dfi_cs_n	MC	DFI chip select
	dfi_odt	dfi_odt	MC	DFI on-die termination control bus
	dif_ras_n	dif_ras_n	MC	DFI row address strobe bus (DDR3 only)
	dfi_reset_n	dfi_reset_n	MC	DFI chip reset
	dfi_we_n	dfi_we_n	MC	DFI write enable signal (DDR3 only)
Write Data Siganls	dfi_wrdata	dfi_wrdata	MC	DFI write data
	dfi_wrdata_cs_n	NA	MC	DFI Write Data Chip Select
	dfi_wrdata_en	dfi_wrdata_en	MC	DFI write enable signal
	dfi_wrdata_mask	dfi_wrdata_mask	MC	DFI Write data byte mask
Read Data Signals	dfi_rddata	dfi_rddata	PHY	dfi interface data
	dfi_rddata_cs_n	NA	MC	DFI Read Data Chip Select
	dfi_rdata_dbi_n	NA	PHY	Read Data DBI
	dfi_rddata_en	NA	MC	Read data enable

Interface Signal Class	Standard DFI	Simplified DFI	Source	Description
	dfi_rddata_valid	dfi_rddata_valid	PHY	The read data of the dfi interface is valid
	dfi_rddata_dnv	NA	PHY	DFI data not valid
Update Signals	dfi_ctrlupd_ack	NA	PHY	MC-initiated update acknowledge
	dfi_ctrlupd_req	NA	MC	MC-initiated update request
	dfi_phyupd_ack	dfi_phyupd_ack	MC	PHY-initiated update acknowledge
	dfi_phyupd_req	dfi_phyupd_req	PHY	PHY-initiated update request
	dfi_phyupd_type	NA	PHY	PHY-initiated update select
Status Signals	dfi_alert_n	NA	PHY	CRC or parity error indicator
	dfi_data_byte_disable	NA	MC	Data byte disable
	dfi_dram_clk_disable	NA	MC	DRAM clock disable
	dfi_freq_ratio	NA	MC	DFI frequency ratio indicator
	dfi_init_complete	dfi_init_complete	PHY	PHY initialization complete
	dfi_init_start	NA	MC	DFI setup stabilization or frequency change initiation
	dfi_parity_in	NA	MC	Parity value
DFI Training Signals	dfi_calvl_capture	NA	MC	CA training capture
	dfi_phy_calvl_cs_n	NA	PHY	CA training chip select
	dfi_calvl_en	NA	MC	PHY CA training logic enable
	dfi_calvl_req	NA	PHY	PHY-initiated CA training request
	dfi_calvl_resp	NA	PHY	CA training response
	dfi_lvl_pattern	NA	MC	Training pattern
	dfi_lvl_periodic	NA	MC	Training length indicator (full or periodic)
	dfi_phylvl_ack_cs_n	NA	MC	DFI PHY training chip select acknowledge
	dfi_phylvl_req_cs_n	NA	PHY	DFI PHY training chip select request
	dfi_phy_rdlvl_cs_n	NA	PHY	Read training chip select for read data eye training
	dfi_phy_rdlvl_gate_cs_n	NA	PHY	Read training chip select for gate training
	dfi_phy_wrlvl_cs_n	NA	PHY	Write leveling chip select
	dfi_rdlvl_en	NA	MC	PHY read data eye training logic enable
	dfi_rdlvl_gate_en	NA	MC	PHY gate training logic enable
	dfi_rdlvl_gate_req	NA	PHY	PHY-initiated gate training request
	dfi_rdlvl_req	NA	PHY	PHY-initiated read data eye training request
	dfi_rdlvl_resp	NA	PHY	Read training response
	dfi_wrlvl_en	NA	MC	PHY write leveling logic enable
	dfi_wrlvl_req	NA	PHY	PHY write leveling request
	dfi_wrlvl_resp	NA	PHY	Write leveling response
	dfi_wrlvl_strobe	NA	MC	Write leveling strobe

Interface Signal Class	Standard DFI	Simplified DFI	Source	Description
Low Power Control Signals	dfi_lp_ack	NA	PHY	Low power acknowledge
	dfi_lp_ctrl_req	NA	MC	Low power opportunity control request
	dfi_lp_data_req	NA	MC	Low power opportunity data request
	dfi_lp_wakeup	NA	MC	Low power wakeup time
Error Signals	dif_error	dif_error	PHY	DFI Error
	dfi_error_info	NA	PHY	DFI Error Info

2. DFI Interface Timing

The working clock frequency ratio of MC and PHY is 1:4, that is, each MC command corresponds to 4 Phase PHY commands. Instructions and data can be distributed among the 4 Phases of the PHY according to requirements. When users send control commands on the DFI interface, they must strictly follow the requirements of DDR SDRAM on command and data delays. DFI commands can only be received after dfi_init_complete is pulled high, and the PHY aligns the data of the read operation.

(1) DFI Interface WriteTiming

To write data to DDR3 SDRAM, you need to send the write command first, and then send the data. The typical timing is shown in Figure 2-25.

- Write operation control: dfi_cs_n, dfi_ras_n, dfi_cas_n, dfi_we_n, dfi_cke, dfi_odt.
- Address line for write operation: dfi_bank, dfi_address.
- Data lines for write operations: dfi_wrdata_en, dfi_wrdata, dfi_wrdata_mask

(2) DFI Interface Read Timing

To read data from DDR3 SDRAM, you need to send the read command first, and then receive the data returned by PHY through DFI. The typical timing is shown in Figure 2-26.

- Control lines for read operations: dfi_cs_n, dfi_ras_n, dfi_cas_n, dfi_we_n, dfi_cke, dfi_odt.
- Address lines for read operations: dfi_bank, dfi_address.
- Data lines for read operations: dfi_rddata, dfi_rddata_valid.

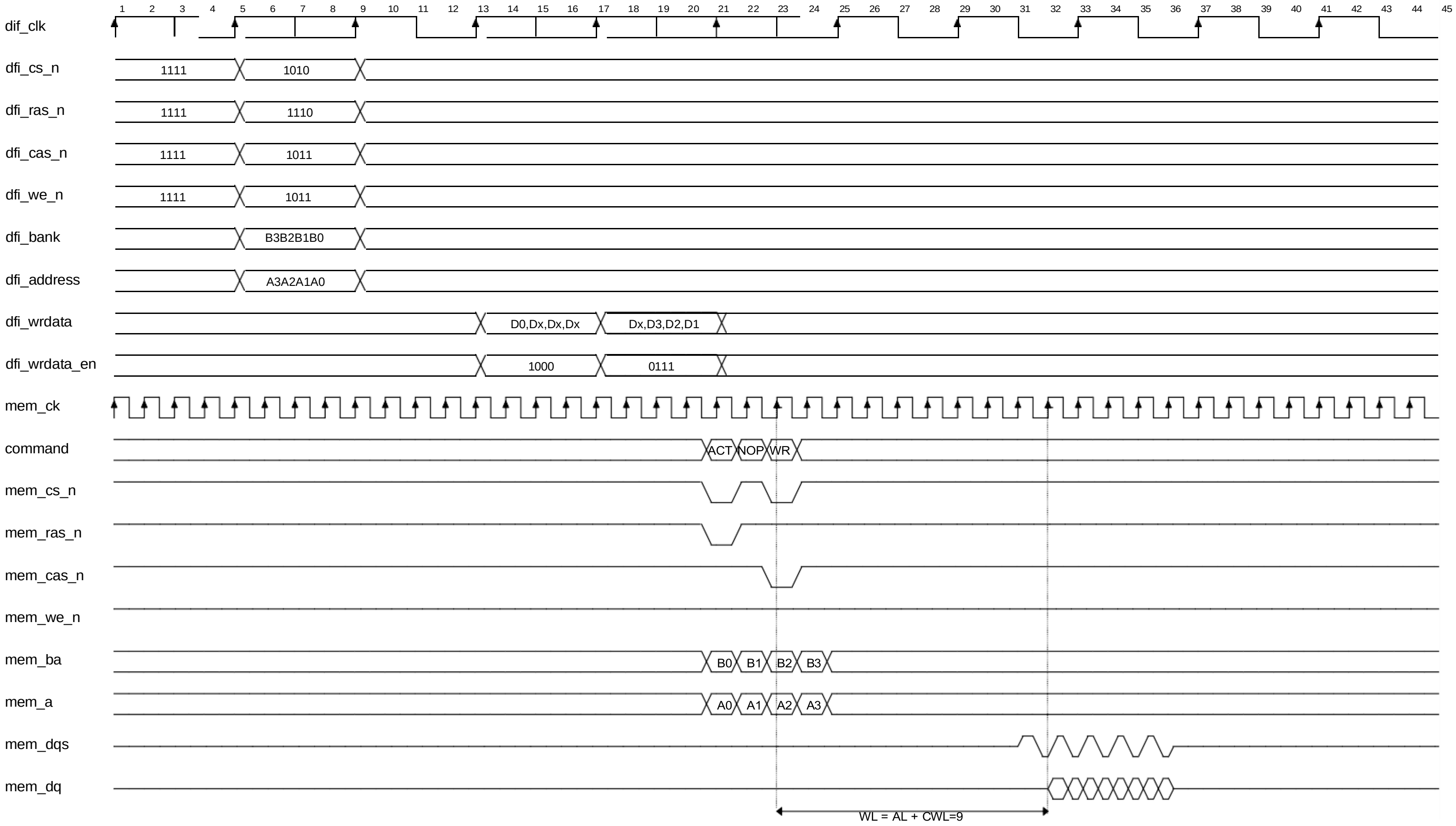


Figure 2-25 DFI interface write operation sequence

Note: This timing diagram is the simulation timing diagram when WL=9.

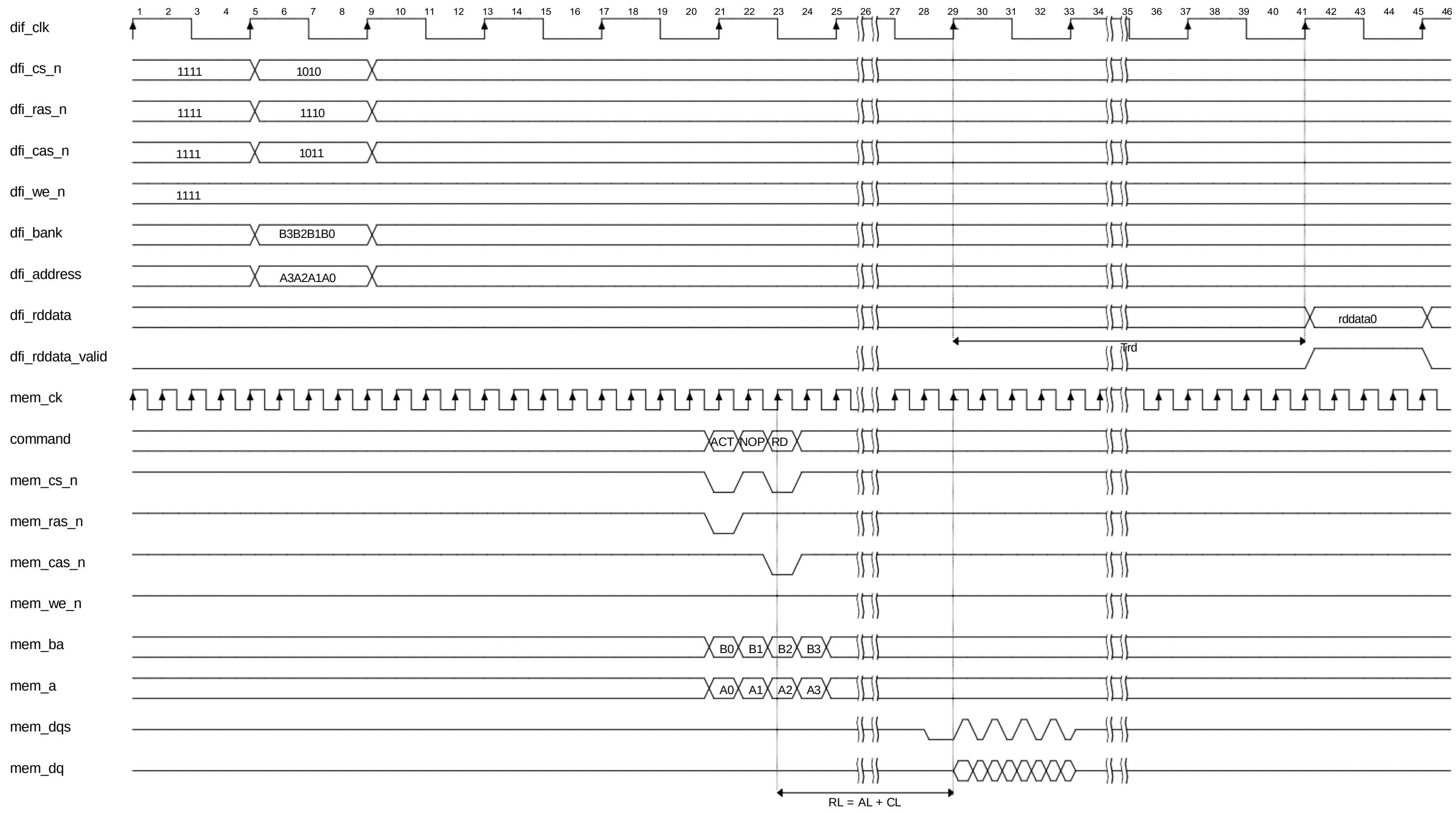


Figure 2-26 DFI interface read operation sequence

Note: This timing diagram is a simulation timing diagram, and Trd is 7 dfi_clk cycles. In fact, the board timing is different from the above diagram.

2.5.5 Debug Data description

1. debug_data description

For the description of each field of debug_data, see Table 2-20.

Table 2-20 Definition of each field of debug_data

Bits	Corresponding Debug port	Description
3:0	coarse_slip_step	Coarse adjustment position obtained by dqs gate training
6:4	read_clk_ctrl	The fine-tuned position obtained by dqs gate training
10:7	gate_win_size	The number of valid windows for dqs gate training
11	gate_check_pass	dqs gate training completion indication (will be set to 0 after completion)
12	rddata_check_pass	Read the calibration data to confirm the completion indication (it will be set to 0 after completion)
20:13	dqs_even_bin	Read the dqs rising edge delay step value obtained after the eye diagram calibration is completed
28:21	dqs_odd_bin	Read the dqs falling edge delay step value obtained after the eye diagram calibration is completed
36:29	total_margin_even	margin of dqs rising edge window
45:38	total_margin_odd	margin of dqs falling edge window
55:48	wrlvl_step	The dqs delay step obtained when write leveling is completed
56	wl_p_ov	Write direction write leveling delay step plus 90 degree code overflow flag 1: overflow 0 does not overflow
57	wrlvl_dq	The dq value returned when writing leveling
65:58	wl_p_dll_bin	write leveling delay code plus 90 degree code

2. debug_calib_ctrl description

For the description of each field of debug_calib_ctrl, see Table 2-21.

Table 2-21 Definition of each field of debug_calib_ctrl

Bits	Corresponding Debug port	Description
0	calib_error	training error indication (generated by write leveling read calibration error)
4:1	dbg_upcal	update control state machine
9:5	dbg_eyecal	Reading Eye Diagram Calibration Control State Machine
14:10	dbg_wrcal	Write Calibration Control State Machine
18:15	dbg_rdcal	MPR Read Calibration Control State Machine
22:19	dbg_wrlvl	write leveling calibration control state machine
26:23	dbg_init	Power-on initialization process control state machine
30:27	dbg_main	training main state machine
34:31	error_status	training error status record: 0: training is correct

Bits	Corresponding Debug port	Description
		1: write leveling training error 2: dqs gate training error 3: read vref training error type 1 4: read vref training error type 2 5: read training error 6: write vref training error 7: write training error 8: data mask (write dbi) training error 9: read vref fine training error 10: read dbi training error 11: eye read training error 12: eye write training error
35	phy_1_calib_en	The PHY that is being trained at the time of training error When instantiated as Ping-Pong PHY: 0: training PHY0 1: training PHY1 Always 0 when instantiated as a single PHY

3. debug_slice_status description

For the description of each field of debug_slice_status, see Table 2-22.

Table 2-22 Definition of each field of debug_slice_status

Bits	Corresponding Debug port	Description
1:0	dll_update_code_don e_tmp	dll update complete indication signal Each bit corresponds to a group, and the high bit corresponds to the high group 1: Done 0: not completed
11:10	dll_lock_tmp	dll lock indication Each bit corresponds to a group (high bit corresponds to high group) 1: locked 0: unlocked
13:12	dqs_gate_comp_done_tmp	dqs gate update complete indication signal Each bit corresponds to a group (a high bit corresponds to a high group) 1: Done 0: not completed
17:16	gate_cal_error_tmp	dqs gate error indication signal Each bit corresponds to a group (a high bit corresponds to a high group) 1: dqs gate error 0: dqs gate no error

Bits	Corresponding Debug port	Description
21:20	gate_check_pass_tm p	dqs gate complete indication signal Each bit corresponds to a group (a high bit corresponds to a high group) 1: dqs gate completed 0: dqs gate not completed
23:22	rdel_move_done_dsli ce	Read calibration single delay adjustment complete indication signal Each bit corresponds to a group (a high bit corresponds to a high group) 1: Adjustment completed 0: adjustment not completed
25:24	rdel_calib_error_dsli ce	Read calibration error indication signal Each bit corresponds to a group (high bit corresponds to high group) 1: Calibration error 0: no error
27:26	rdel_calib_done_dsli ce	Read calibration complete indication signal Each bit corresponds to a group (a high bit corresponds to a high group) 1: Calibration completed 0: not completed
31:30	wrlvl_dqs_resp_tmp	write leveling calibration complete indication signal Each bit corresponds to a group (a high bit corresponds to a high group) 1: Calibration completed 0: not completed
33:32	wrlvl_error_tmp	write leveling calibration error indication signal Each bit corresponds to a group (high bit corresponds to high group) 1: Calibration error 0: no error

4. debug_slice_state description

For the description of each field of debug_slice_state, see Table 2-23.

Table 2-23 Definition of each field of debug_slice_state

Bits	Corresponding Debug port	Description
7:0	dbg_slice_rdchk_state	Read Calibration Data Compare State Machine Status Output
11:8	dbg_slice_rpbd_ctrl_state	Read Calibration Control dqs Delayed State Machine Status Output
14:12	dbg_slice_gate_state	Adjust the state output of the dqs gate delay state machine
18:15	dbg_slice_dq_wrcal_state[3:0]	Control write direction dq delay state machine state output
21:19	dbg_slice_wrlvl_state	Control write leveling delay state machine state output

2.6 IP Register description

This section introduces the description and access methods of HMIC_S IP related registers.

2.6.1 Register description

1. Controller Internal register

(1) MODE_REG_0_ADDR

The definition of this register in DDR3/DDR4 mode is the same as MR0 in DDR3 SDRAM Standard[9]/DDR4 SDRAM Standard[10], the bit width is 16bit, and the access address is 0x00.

Table 2-24 Bit definition of DDR3 MODE_REG_0_ADDR

Bits	Name	Reset value	Access type	Description
1:0	BL	2’b00	R	Burst Length
2	CL[0]	Instantiated value	R/W	CAS Latency (CL 0th bit)
3	RBT	Instantiated value	R/W	Read Burst Type
6:4	CL[3:1]	Instantiated value	R/W	CAS Latency (CL [3:1] bit) CL max is 8
7	TM	1’b0	R	Mode
8	DLL	0	R	DLL Reset
11:9	WR	Instantiated value	R/W	Write recovery for autoprecharge
12	PPD	1	R	DLL Control for Precharge PD
15:13	Reserve	0	R	Reserve

Table 2-25 Bit definition of DDR4 MODE_REG_0_ADDR

Bits	Name	Reset value	Access type	Description
1:0	BL	2’b00	R	Burst Length
2	CL[0]	Instantiated value	R/W	CAS Latency (CL 0th bit)
3	RBT	Instantiated value	R/W	Read Burst Type
6:4	CL[3:1]	Instantiated value	R/W	CAS Latency (CL [3:1] bit) CL max is 8
7	TM	1’b0	R	Mode
8	DLL	0	R	DLL Reset
11:9	WR_RTP	Instantiated value	R/W	Write Recovery and Read to Precharge for auto precharge
15:12	Reserve	0	R	Reserve

(2) MODE_REG_1_ADDR

The definition of this register in DDR3/DDR4 mode is the same as MR1 in DDR3 SDRAM Standard[9]/DDR4 SDRAM Standard[10], the bit width is 16bit, and the access address is 0x01.

Table 2-26 Bit definition of DDR3 MODE_REG_1_ADDR

Bits	Name	Reset value	Access type	Description
0	DLL	1’b0	R	DLL Enable
1	D.I.C[0]	1’b0	R/W	Output Driver Impedance Control (D.I.C bit 0)
2	Rtt_Nom[0]	Instantiated value	R/W	Rtt_Nom bit 0
4:3	AL	2’b10	R	Additive Latency
5	D.I.C[1]	Instantiated value	R/W	Output Driver Impedance Control (D.I.C bit 1)
6	Rtt_Nom[1]	1’b0	R/W	Rtt_Nom 1st bit
7	Level	1’b0	R	Write leveling enable[12]
8	Reserve	1’b0	R	Reserve
9	Rtt_Nom[2]	Instantiated value	R/W	Rtt_Nom bit 2
10	Reserve	1’b0	R	Reserve
11	TDQS	1’b0	R	TDQS enable
12	Qoff	1’b0	R	Qoff
15:13	Reserve	3’b000	R	Reserve

[12] See DDR3 SDRAM Standard[9] for details.

Table 2-27 Bit definition of DDR4 MODE_REG_61_ADDR

Bits	Name	Reset value	Access type	Description
0	DLL	1’b0	R	DLL Enable
2:1	D.I.C	2’b0	R/W	Output Driver Impedance Control
4:3	AL	2’b00	R	Additive Latency
6:5	Reserve	2’b0	R	Reserve
7	Level	1’b0	R	Write leveling enable[12]
10:8	Rtt_Nom	Instantiated value	R/W	Rtt_Nom
11	TDQS	1’b0	R	TDQS enable
12	Qoff	1’b0	R	Qoff
15:13	Reserve	3’b000	R	Reserve

(3) MODE_REG_2_ADDR

The definition of this register in DDR3/DDR4 mode is the same as MR2 in DDR3 SDRAM Standard[9]/DDR4 SDRAM Standard[10], the bit width is 16bit, and the access address is 0x02.

Table 2-28 Bit definition of DDR3 MODE_REG_2_ADDR

Bits	Name	Reset value	Access type	Description
2:0	PASR	3’b0	R	Partial Array Self-Refresh (Optional)
5:3	CWL	Instantiated value	R/W	CAS write Latency
6	ASR	1’b0	R	Auto Self-Refresh
7	SRT	1’b0	R	Self-Refresh Temperature Rang
8	Reserve	1’b0	R	Reserve
10:9	Rtt_WR	2’b00	R/W	Rtt_WR
15:11	Reserve	5’b0	R	Reserve

Table 2-29 Bit definition of DDR4 MODE_REG_2_ADDR

Bits	Name	Reset value	Access type	Description
2:0	Reserve	3’b0	R	Reserve
5:3	CWL	Instantiated value	R/W	CAS write Latency
7:6	LP ASR	1’b0	R	LoW Auto Self-Refresh
7	SRT	1’b0	R	Self-Refresh Temperature Rang
8	Reserve	1’b0	R	Reserve
10:9	Rtt_WR	2’b00	R/W	Rtt_WR
11	Reserve	1’b0	R	Reserve
12	WCRC	1’b0	R	Write CRC
15:11	Reserve	5’b0	R	Reserve

(4) MODE_REG_3_ADDR

The definition of this register in DDR3/DDR4 mode is the same as MR3 in DDR3 SDRAM Standard[9]/DDR4 SDRAM Standard[10], the bit width is 16bit, and the access address is 0x03.

Table 2-30 Bit definition of DDR3 MODE_REG_3_ADDR

Bits	Name	Reset value	Access type	Description
[15:0]	Reserve	16’h0	R	Reserve[13]

Table 2-31 Bit definition of DDR4 MODE_REG_3_ADDR

Bits	Name	Reset value	Access type	Description
1:0	MPR_PA	2’b0	R/W	MPR page Selection
2	MPR_OP	1’b0	R/W	MPR Operation
3	GM	1’b0	R/W	Geardown Mode
4	PDA	1’b0	R/W	Per DRAM Addressability
5	TSR	1’b0	R/W	Temperature sensor readout
8:6	FGRM	3’b0	R/W	Fine Granularity Refresh Mode
10:9	WCLCD	2’b00	R/W	Write CMD Latency when CRC and DM are enabled
12:11	MPR_RF	2’b00	R/W	MPR Read Format
15:13	Reserve	3’b0	R	Reserve

[13] See DDR3 SDRAM Standard[9] for details

(5) MODE_REG_4_ADDR

This register is only used in DDR4 mode, and its definition is the same as MR4 in DDR4 SDRAM Standard[10], the bit width is 16bit, and the access address is 0x04.

Table 2-32 Bit definition of DDR4 MODE_REG_4_ADDR

Bits	Name	Reset value	Access type	Description
0	Reserve	1’b0	R	Reserve
1	MAX_PDM	1’b0	R/W	Maximum Power Down Mode
2	TCRR	1’b0	R/W	Temperature Controlled Refresh Range
3	TCRM	1’b0	R/W	Temperature Controlled Refresh Mode
4	IVM	1’b0	R/W	Internal Vref Monitor
5	Reserve	1’b0	R	Reserve
8:6	CCLM	3’b0	R/W	CS to CMD/ADDR Latency Mode (cycles)
9	SRA	1’b0	R/W	Self Refresh Abort
10	RPTM	1’b0	R/W	Read Preamble Training Mode
11	RP	1’b0	R/W	Read Preamble
12	WP	1’b0	R/W	Write Preamble
15:13	Reserve	3’b0	R	Reserve

(6) MODE_REG_5_ADDR

This register is only used in DDR4 mode, and its definition is the same as MR5 in DDR4 SDRAM Standard[10], the bit width is 16bit, and the access address is 0x05.

Table 2-33 Bit definition of DDR4 MODE_REG_5_ADDR

Bits	Name	Reset value	Access type	Description
2:0	CPLM	3’b0	R/W	C/A Parity Latency Mode
3	CEC	1’b0	R/W	CRC Error Clear
4	CPES	1’b0	R/W	C/A Parity Error Status
5	ODT_IB	1’b0	R/W	ODT Input Buffer during Power Down mode
8:6	RTT_PARK	1’b0	R/W	RTT_PARK
9	CAPE	1’b0	R	CA parity Persistent Error
10	DM	1’b0	R/W	Data Mask
11	WDBI	1’b0	R/W	Write DBI
12	RDBI	1’b0	R/W	Read DBI
15:13	Reserve	3’b0	R	Reserve

(7) MODE_REG_6_ADDR

This register is only used in DDR4 mode, its definition is the same as MR6 in DDR4 SDRAM Standard[10], the bit width is 16bit, and the access address is 0x06.

Table 2-34 Bit definition of DDR4 MODE_REG_6_ADDR

Bits	Name	Reset value	Access type	Description
5:0	VREFTV	6’b0	R/W	VrefDQ Training Value
6	VREFTR	1’b0	R/W	VrefDQ Training Range
7	VREFTE	1’b0	R/W	VrefDQ Training Enable
9:8	Reserve	3’b0	R	Reserve
12:10	CCD_L	3’b0	R/W	tCCD_L
15:13	Reserve	3’b0	R	Reserve

(8) CTRL_MODE_DATA

This register is used to store the instruction (MRS/Low Power/Normal) issued by the user, and the access address is 0x04. See Table 2-35 for the definition of each bit.

Table 2-35 Bit definitions of CTRL_MODE_DATA

Bits	Name	Reset value	Access type	Description
0	enable	1’b0	R/W	ddr core state switching trigger enable
13:1	Reserve	13’h0	R	Reserve
15:14	select	2’b00	R/W	DDR Core state selection 00: Normal 01: MRS 10: Self-Refresh 11: Power Down

(9) STATUS_REG_DATA

This register is used to store the current state of DDR, and the access address is 0x05. For the definition of each bit, see Table 2-36.

Table 2-36 Bit definitions of STATUS_REG_DATA

Bits	Name	Reset value	Access type	Description
1:0	mode_state	2’b0	R	Store the current state of DDR3/DDR4 2’b00: Normal State 2’b01: Self-Refresh State 2’b10: Power Down State 2’b11:MRS State
14:2	Reserve	13’b0	R	Reserve
15	busy	1’b0	R	Whether the current request of DDR3/DDR4 is responded 1’b0: The current request has been responded to 1'b1: The current request is not being processed

2. PHY Interface registers

The internal registers of DDR PHY support DDR3 and DDR4, and the meaning is completely consistent with the protocol. For details, please refer to JESD79-3D, DDR3 SDRAM Standard[9], JESD79-4B, DDR4 SDRAM Standard[10].

2.6.2 Register access

The internal registers of the Controller are accessed through the APB interface. The register data is passed through apb_wdata and apb_rdata through the address, and the register address is passed through apb_addr. For the address value of each register, please refer to "2.6.1 Register Description".

The access method of PHY internal register is the same as that of accessing Memory internal Mode Register register.

2.7 Typical application

2.7.1 Single BANK-x16 typical application

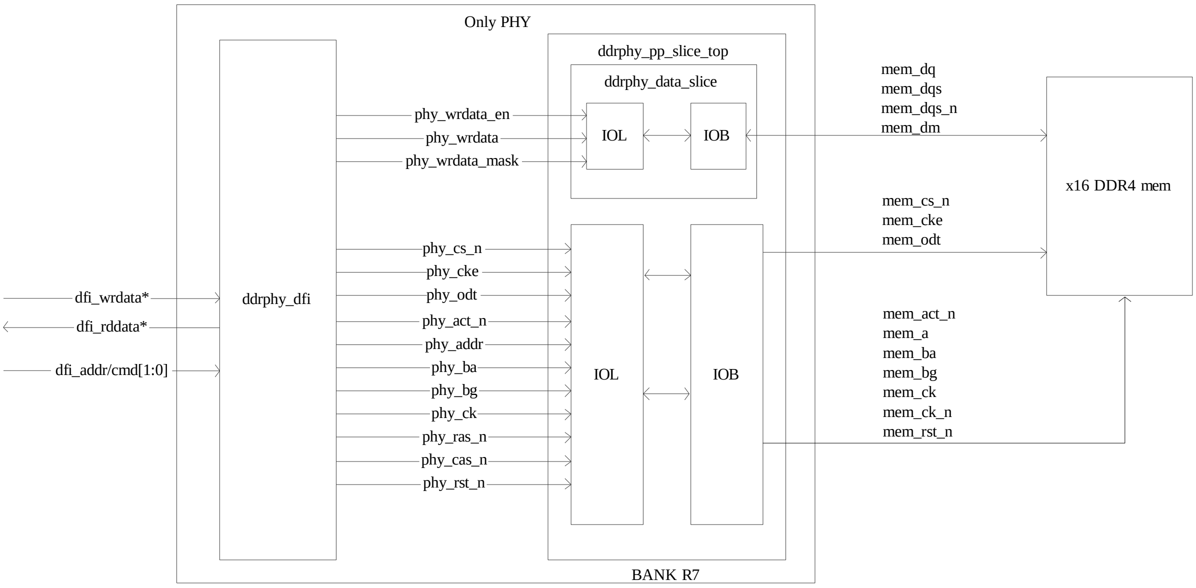


Figure 2-27 Schematic diagram of single BANK-x16 structure

A typical application for a single BANK-x16 is shown in Figure 2-27. Single BANK-x16 is based on a certain HP BANK (R7 in the figure), and saves pins by removing some functional pins. Single BANK-x16 application has the following considerations:

- When the HARD FIFO function is enabled, only CKE is allowed to be placed on the dedicated calibration resistor pins (there are two dedicated calibration resistor pins in the HP BANK).
- Functional pins (CS_n, TEN, PARITY, ALERT_n) can be configured through the configuration interface, see "2.3.1 2(3) Step 3: Pin/Bank Options" for the configuration method.

2.7.2 Dual BANK-x32 typical application

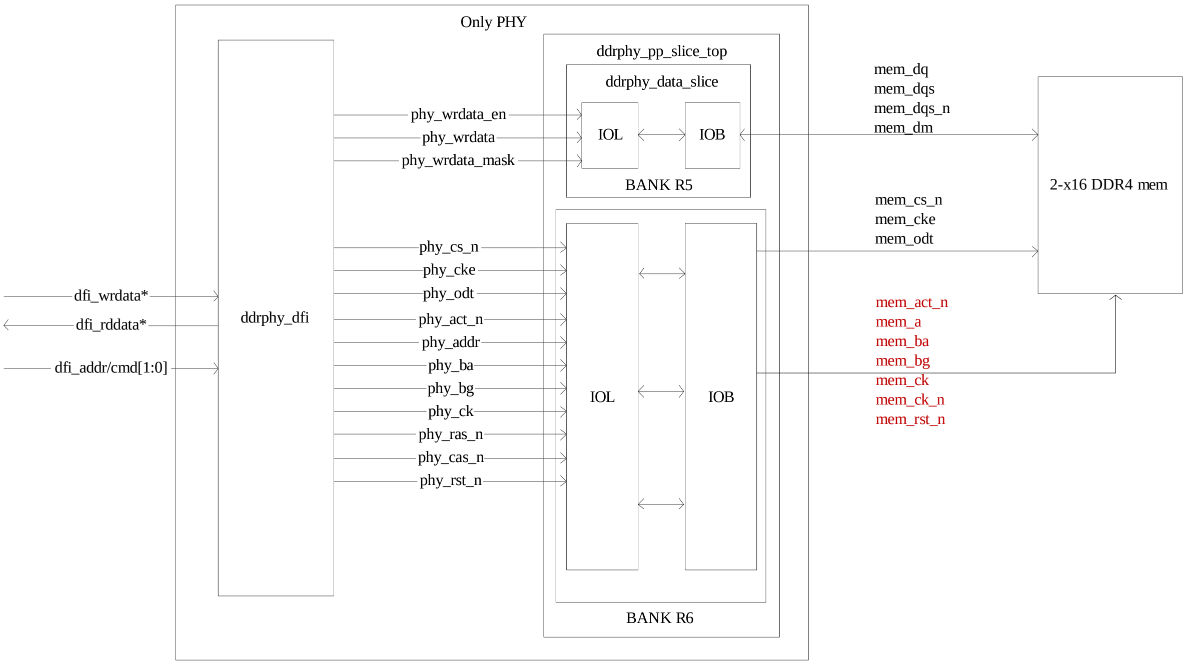


Figure 2-28 Schematic diagram of dual BANK-x32 structure

A typical application for dual BANK-x32 is shown in Figure 2-28. Dual BANK-x32 is based on two adjacent HP BANKs (R5, R6 in the figure), among which BANK R6 is used for CA channel, and the other BANK R5 is used for DQ channel. Dual BANK-x32 applications have the following considerations:

- Do not use the two dedicated calibration resistor pins in the HP BANK.
- It is forbidden to distribute signals across banks. For specific Pin & Bank design rules, please refer to "UG050010_Titan2 Series HMIC_S IP Application Guide" [12].

2.7.3 Multy BANK-x64 typical application

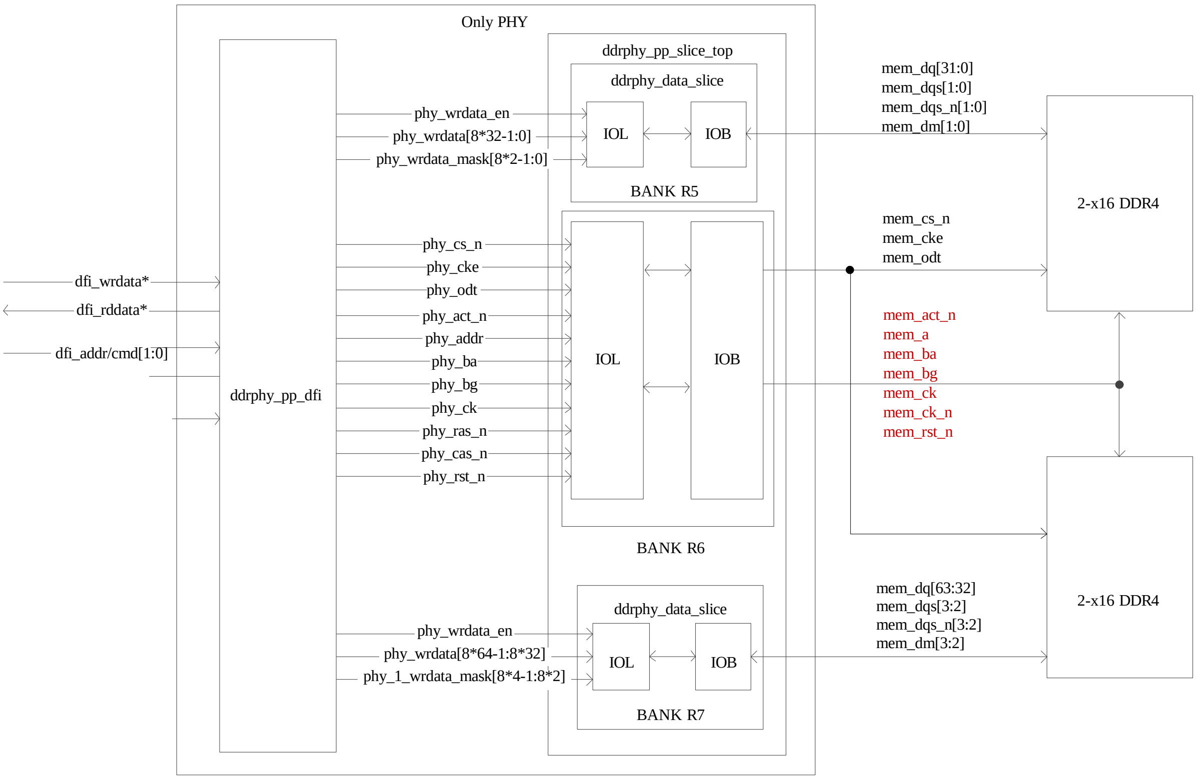


Figure 2-29 Schematic diagram of multi-bank-x64 structure

A typical application for multi-bank-x64 is shown in Figure 2-29. Multi-BANK-x64 is based on two adjacent HP BANKs (R5, R6, R7 in the figure), of which BANK R6 is used for the CA channel, and the other BANK R5 and BANK R7 are used for the DQ channel. Multiple BANK-x64 applications have the following considerations:

- Do not use the two dedicated calibration resistor pins in the HP BANK.
- It is recommended that the CA signal be distributed on BANK R6 to ensure that the delay difference of the clock distribution path is relatively short. For specific Pin & Bank design rules, please refer to "UG050010_Titan2 Series HMIC_S IP Application Guide" [12].
- Under the Fly-by topology, ensure that the remote particle stub routing meets the design requirements as much as possible to ensure that the AC parameters can be covered.

2.7.4 Ping Pong PHY typical application

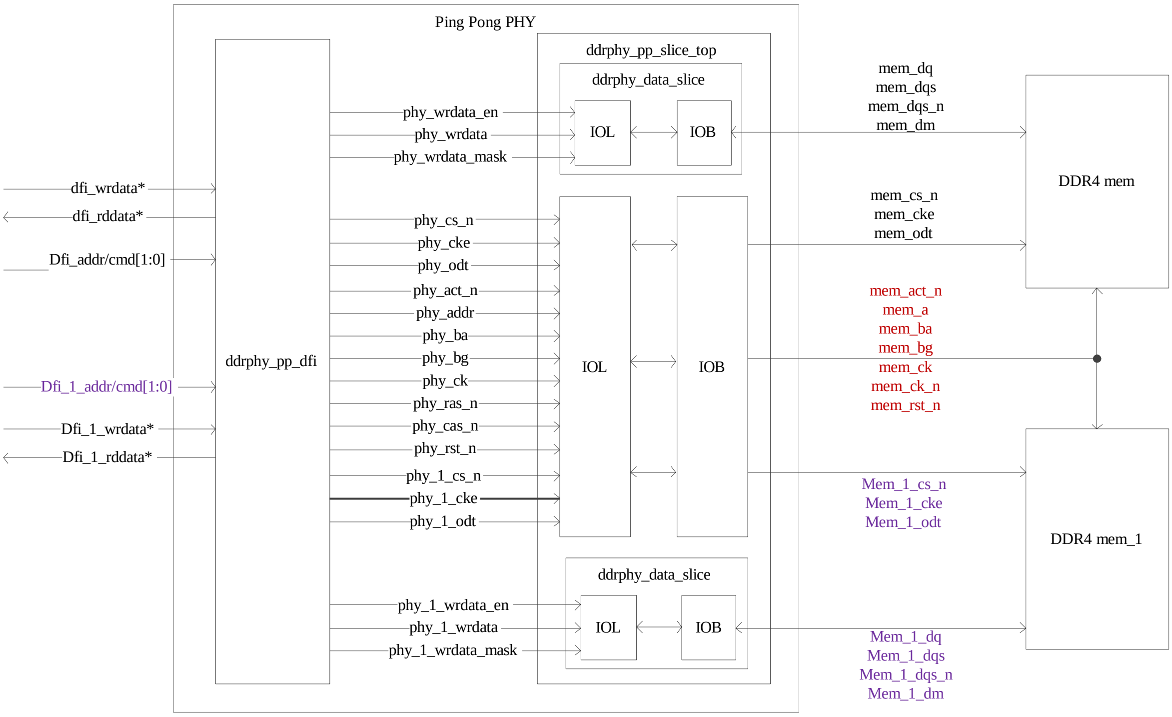


Figure 2-30 Schematic diagram of Ping Pong PHY structure

A typical application for Ping Pong PHY is shown in Figure 2-30. The Ping Pong PHY saves pins by sharing (mostly) the addr/cmd pins of both PHYs. The Ping Pong PHY application has the following considerations.

- In order to synchronize with the original design, Ping Pong PHY retains the original interface naming rules, and only adds *_1* to the newly added PHY for distinction.
- The two DDR mems must be the same Component, and the parameter settings must be the same.
- On the DFI interface, the original four time slots of each cycle are allocated according to the following rules: time slot 0/2 belongs to DDR4 mem, and time slot 1/3 belongs to DDR4 mem_1.
- Complete the training of DDR mem first, and then complete the training of DDR mem_1.

2.8 Instruction and Precautions

2.8.1 AXI4 Interface Burst Computing

Start_Address = ADDR;

Burst_Length= len;

End_Address = ADDR + len * 8.

2.8.2 Clock constraints

HMIC_S IP has multiple clocks that need to be constrained, namely ref_clk, rst_clk, ddrphy_sysclk, and phy_dq_sysclk (one for each BANK needs to be constrained), and the relationship between each clock is shown in Figure 2-31. For the specific constraint method, please refer to the .fdc file in the IP /pnr directory.

Among them, ref_clk is the input reference clock, ddrphy_sysclk and rst_clk are obtained by multiplying the frequency of GPLL, and phy_dq_sysclk is obtained by multiplying the frequency of PPLL. ddrphy_sysclk is the system clock of IP soft logic, rst_clk is the driving clock of GPLL dynamic phase shift and reset sequence, phy_dq_sysclk is the BANK clock, each BANK has a PPLL, and the phy_dq_sysclk of the BANK is obtained by frequency division of the PPLL of the BANK.

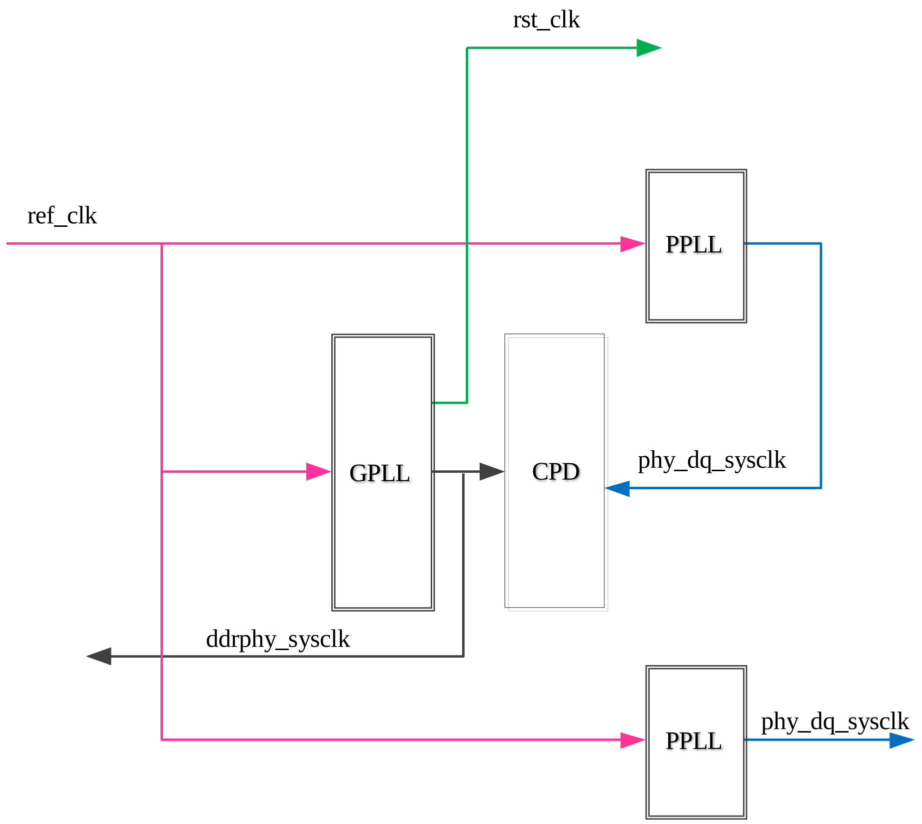


Figure 2-31 IP internal clock structure

2.8.3 IO Constraints

The position constraints on the internal modules of the chip need to be consistent with the position constraints provided by the .fdc file. The constraint positions cannot be changed, and the path can be modified according to the specific usage.

Constraints on parameters such as pin positions and level standards are carried out in UCE (User Constraint Editor). For specific parameter configuration, please refer to the .fdc file in the IP/pnr directory. Some requirements for pin parameter settings are as follows.

- VCCIO needs to be set to 1.5V or 1.2V (DDR3 is 1.5, DDR4 is 1.2);
- mem_ck is a differential output signal, which needs to use SSTL15D_I or POD12D level;
- mem_dqs is a bidirectional differential signal, which needs to use SSTL15D_I or POD12D level;
- The remaining signals are single-ended signals, which need to use SSTL15_I and POD12 levels;
- mem_dq is a bidirectional single-ended signal, it needs to configure the reference voltage VREF, it is recommended to use an external VREF, select VREF_MODE as OUT, and then select the corresponding external VREF input pin;
- The mem_dqs/mem_dq signal needs to be matched by a terminal resistor on the FPGA side. It is recommended to use an internal terminal resistor, set DDR_TERM_MODE to ON, and set CP_DYN_TERM to ON;
- The rest of the parameters can be set by default.

2.8.4 Physical constraints

The DDR project requires physical constraints, which can be edited directly through fdc, or manually constrained through the PCE (Physical Constraint Editor) tool. Physical constraint information is stored in .fdc files in the /pnr directory and is eventually passed to the pcf constraint file during layout. For the specific configuration method of the constraint file, please refer to the help document under the PDS installation path: User_Constraint_Editor_User_Guide[5], Physical_Constraint_Editor_User_Guide[6]. For example:

```
define_attribute {i:I_ips_ddr_top.u_ddrphy_top.u_ddrphy_cpd} {PAP_LOC}
{DDRPHY_CPD_HP_669_310}

define_attribute {i:I_ips_ddr_top.u_ddrphy_top.ddrphy_gpll.u_gpll} {PAP_LOC}
{GPLL_667_463}
```

2.8.5 dqs gate update function description

When the temperature changes, the dqs phase in the read direction may change, causing the dqs gate window to shift, resulting in dqs gate errors, so it is necessary to perform dqs gate update operations regularly.

1. Timing description

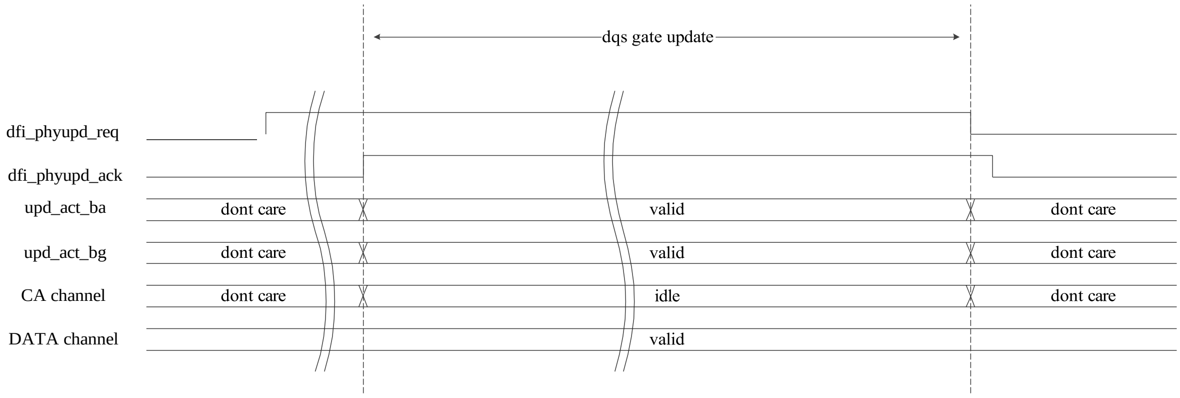


Figure 2-32 Schematic diagram of dqs gate update

When the dqs gate needs to update, it will pull up the dfi_phyupd_req signal and send a request to the controller. When the controller is ready to respond, it will pull up the dfi_phyupd_ack signal. At this time, ddrphy starts to execute the dqs gate update operation. After the update is completed, pull down the dfi_phyupd_req, and the controller detects dfi_phyupd_req Pull low after pull low dri_phyupd_ack.

During the period when dfi_phyupd_ack is pulled high to dfi_phyupd_req is pulled low, upd_act_ba and upd_act_bg need to remain valid, the ca bus of dfi cannot send commands, and the read and write data bus of dfi is not restricted by dqs gate update, and ddrphy will automatically shield the read data during dqs gate update.

2. Working mode

The dqs gate update process needs to adjust the phase of the dqs gate, and then read the Memory data to determine whether the dqs gate is correct to determine whether the phase of the dqs gate needs to be adjusted. Considering that dqs gate update will occupy normal read and write bandwidth, dqs gate update has designed two working modes.

Description:

Users can choose the working mode of dqs gate update according to the actual Memory operation process to improve the efficiency of Memory operation.

---

(1) Mode 1

Before dfi_phyupd_ack is pulled high, to make Memory enter the Idle state, ddrphy first performs the Active operation, upd_act_ba and upd_act_bg decide which Bank is Active. By default, Active is row 0 of the selected Bank, then dqs gate update is performed, and finally

Precharge is executed.

(2) Mode 2

Controller can pull up dfi_phyupd_ack during reading and writing, tell ddrphy which Bank is currently Active through upd_act_ba and upd_act_bg, ddrphy directly executes dqs gate update, and then pulls down dfi_phyupd_req.

3. Mode selection

Both modes are selected by parameter GATE_ACT_PRE_EN.

- GATE_ACT_PRE_EN = 1 select mode 1;
- GATE_ACT_PRE_EN = 0 select mode 2.

2.9 IP Debuggin means

Description:

For more information about IP debugging, please refer to "UG050010_Titan2 Series HMIC_S IP Application Guide" [12].

2.9.1 Key indication signals

For some key information during the operation of DDRPHY, a single-bit indicator signal is made for easy observation, which can be connected to an external LED or monitored in other ways, and can quickly judge the operating status of DDRPHY. See Table 2-37 for key indication signal descriptions.

Table 2-37 Key Indicator Signals

Signal	I/O	Width	Description
pll_lock	O	1	PLL lock indication, high level means locked.
ddrphy_cpd_lock	O	1	cpd lock indication, high level means locked.
dfi_init_complete	O	1	The PHY has completed the training-related operations and is in the normal state.
heart_beat_led	O	1	heartbeat signal. When the ddrphy system clock is normal, it flashes once in about 1 second. (signal generated in Example Design)

Signal	I/O	Width	Description
dfi_bist_run_led	O	1	Data flow indication signal. When the data flow is normal, the 2^13 beat data flashes once. This signal is stable at 0 or 1 when data flow is interrupted. (signal generated in Example Design)
err_flag_led	O	1	Data check error signal. 1: There is an error in data detection, which needs to be cleared manually after the error is eliminated. 0: No error in data detection. (signal generated in Example Design)

2.9.2 Internal Status and Control Signals

For some other non-real-time changing internal status signals and control signals used for debugging, see Table 2-17. These signals can be easily read and written through the serial port. Flexible debugging can be realized by developing companion scripts.

Refer to "Figure 2-14 Example Design System Block Diagram" for the serial port access schematic diagram. The ips2l_uart_ctrl module in DDRPHY Example Design is a serial port communication module. The host computer realizes the read and write operations of internal status signals and control signals through the serial port and the ips2l_uart_ctrl module.

The serial port configuration adopted by DDRPHY Example Design is shown in Table 2-38.

Table 2-38 DDRPHY Example Design serial port configuration

Baud rate	Start bit	Data bit	Stop bit	Check bit	Flow control
115200bps	1bit	8bit	1bit	None	None

Description:

A set of serial port scripts were developed during the internal debugging process, please contact AE to obtain them if necessary.

Chapter 3 Appendix

3.1 Reference Documentation

[1] Pango_Design_Suite_Quick_Start_Tutorial

[2] Pango_Design_Suite_User_Guide

[3] IP_Compiler_User_Guide

[4] Simulation_User_Guide

[5] User_Constraint_Editor_User_Guide

[6] Physical_Constraint_Editor_User_Guide

[7] Route_Constraint_Editor_User_Guide

[8] UG050007_Titan2 Series GTP User Guide

[9] JESD79-3D, DDR3 SDRAM Standard

[10] JESD79-4B, DDR4 SDRAM Standard

[11] DDR PHY Inteface, DFI 3.1 Specification

[12] UG050010_Titan2 Series HMIC_S IP Application Guide

3.2 Glossary

A

APB Advanced Peripheral Bus

AXI AdvancedeXtensible Interface

D

DDRDouble Data Rate

DFI DDR PHY Interface

L

LP Low Power

M

MC Memory Controller

MR Mode Register

MRS Mode Register Set

P

PHY Physical

R

RD Read

U

UI User Interface

W

WR Write

3.3 List of abbreviations

D

DCD DDR Command Decode

DCP DDR3 Command Procedure

H

HMIC High performance Memory Interface Controller

I

IPC IP Compiler

P

PDS Pango Design Suite

PCE Physical Constraint Editor

U

UCE User Constraint Editor

3.4 Statement

3.4.1 Copyright Notice

3.4.2 Disclaimer

This document only provides phased information, and the content contained in it may be updated at any time according to the actual situation of the product without prior notice. The company does not assume any legal responsibility for any direct or indirect losses caused by improper use of this document.

THIS DOCUMENT IS PROVIDED AS IS, WITHOUT WARRANTY OF ANY KIND, INCLUDING ANY WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT, AND ANY WARRANTY ANY PROPOSAL, SPECIFICATION OR SAMPLE ARE OTHERWISE REFERENCED. No license, express or implied, to any intellectual property right is hereby granted by this document, whether by estoppel or otherwise.

The company reserves the right to modify the relevant documents of the company's series of products at any time without prior notice.

Замечания по переводу отправляйте в ЛС пользователю Zig сайта electronix.ru